ASME
SETTING THE STANDARD



**Transactions of the ASME, Journal of Applied Mechanics** (ISSN 0021-8936) is published bimonthly (Jan., Mar., May, July, Sept., Nov.) by The American Society of Mechanical Engineers, Three Park Avenue, New York, NY 10016. Periodicals postage paid at New York, NY and additional mailing offices. POSTMASTER: Send address changes to Transactions of the ASME, Journal of Applied Mechanics, c/o THE AMERICAN SOCIETY OF MECHANICAL ENGINEERS, 22 Law Drive, Box 2300, Fairfield, NJ 07007-2300. **CHANGES OF ADDRESS** must be received at Society headquarters seven weeks before they are to be effective. Please send old label and new address. **STATEMENT from By-Laws.** The Society shall not be responsible for statements or opinions advanced in papers or printed in its publications (B7.1, Para. 3).


# Journal of Applied Mechanics

Published Bimonthly by ASME

VOLUME 76 • NUMBER 2 • MARCH 2009

## PREFACE



## STABILIZED, MULTISCALE, AND MULTIPHYSICS METHODS IN FLUID MECHANICS



(Contents continued on inside back cover)

This journal is printed on acid-free paper, which exceeds the ANSI Z39.48-1992 specification for permanence of paper and library materials. ♾™
♻ 85% recycled content, including 10% post-consumer fibers.

**Journal of Applied Mechanics** **Volume 76, Number 2** **MARCH 2009**



The ASME Journal of Applied Mechanics is abstracted and indexed in the following:

*Alloys Index, Aluminum Industry Abstracts, Applied Science & Technology Index, Ceramic Abstracts, Chemical Abstracts, Civil Engineering Abstracts, Compendex (The electronic equivalent of Engineering Index), Computer & Information Systems Abstracts, Corrosion Abstracts, Current Contents, EEA (Earthquake Engineering Abstracts Database), Electronics & Communications Abstracts Journal, Engineered Materials Abstracts, Engineering Index, Environmental Engineering Abstracts, Environmental Science and Pollution Management, Fluidex, Fuel & Energy Abstracts, GeoRef, Geotechnical Abstracts, INSPEC, International Aerospace Abstracts, Journal of Ferrocement, Materials Science Citation Index, Mechanical Engineering Abstracts, METADEX (The electronic equivalent of Metals Abstracts and Alloys Index), Metals Abstracts, Nonferrous Metals Alert, Polymers Ceramics Composites Alert, Referativnyi Zhurnal, Science Citation Index, SciSearch (Electronic equivalent of Science Citation Index), Shock and Vibration Digest, Solid State and Superconductivity Abstracts, Steels Alert, Zentralblatt MATH*

Journal of Applied Mechanics

# Preface

## Special Issue on Stabilized, Multiscale, and Multiphysics Methods in Fluid Mechanics

This special issue of the *Journal of Applied Mechanics* is based on the ASME International Mechanical Engineering Congress and Exposition ASME05 and ASME06. The mini-symposium on *Challenges and Advances in Flow Simulation and Modeling: Fundamental and Enabling Technologies* was held at ASME05 in Orlando, Florida, November 5–11, 2005, and the mini-symposium on *Stabilized, Multiscale and Multiphysics Methods* was held at ASME06 in Chicago, Illinois, November 5–10, 2006. The scope of the two symposia included all aspects of the stabilized and multiscale finite element methods as well as their applications to coupled fluid-structure interaction problems. The papers presented at the two symposia included (i) mathematical theory of the stabilized and multiscale finite element methods, (ii) new stabilized formulations, (iii) stabilized methods applied to fluid-structure interaction problems, (iv) large scale computations with stabilized methods, and (v) application of stabilized methods to biofluid dynamics.

This special issue contains 12 papers that present a spectrum of physical problems where stabilized methods have been applied.

The paper by Catabriga, Souza, Coutinho and Tezduyar presents inviscid compressible flow calculations with the *YZB* shock-capturing parameter. The shock-capturing parameter based on conservation variables is compared with a parameter based on the entropy variables, and a sequence of numerical tests including 1D, 2D and 3D examples are presented.

Rispoli, Saavedra, Menichini and Tezduyar present an application of the *YZB* shock capturing technique integrated in a variable subgrid scale formation for inviscid supersonic flows. They show a variety of test problems for high Mach number flows.

Corsini, Menichini, Rispoli, Santoriello and Tezduyar develop a stabilization technique that is based on a variational multiscale method. Their technique includes a discontinuity-capturing term designed to be operative when the solution gradients are high and the reaction-like terms are dominant. They show applications of their method on a sequence of 2D and 3D problems.

The work by Manguoglu, Saied, Sameh, Tezduyar and Sathe presents new preconditioning techniques for solving the nonsymmetric systems that arise from the discretization of the Navier–Stokes equations. They also show the effectiveness of their proposed techniques for handling time-accurate as well as steady-state solutions.

The work by Houseman, Kiris and Hafez presents time-derivative preconditioning methods for numerical simulation of inviscid multicomponent and multiphase flows. Their paper deals with Riemann problems and presents two-dimensional applications. The time-derivative preconditioned system of equations is shown to be hyperbolic in time and well-conditioned in the incompressible limit.

The work by Takagi, Yamada, Gong and Matsumoto presents an application of fluid mechanics to biofluid dynamics and to the deformation of vesicles in a shear flow.

The paper by Cruchaga, Celentano and Tezduyar presents the numerical and experimental analyses of the collapse of a water column over an obstacle. Their computational model is based on a stabilized formulation that is integrated with a moving interface technique, namely the Edge-Tracked Interface Locator Technique (ETILT) to calculate the evolution of the water-air interface.

A vector extrapolation method for strong coupling of the fluid-structure interaction solvers is presented by Kuttler and Wall. They consider the case of an incompressible fluid and nonlinear elastodynamics, and present polynomial based vector extrapolation schemes that are then applied to coupled FSI problems.

Brummelen presents the added mass effects of compressible and incompressible flows in fluid-structure interaction. This work shows that on increasingly small time intervals, the added mass of a compressible flow is proportional to the length of the time interval, whereas the added mass of an incompressible flow approaches a constant. The paper then presents the implications of this difference on the stability and accuracy of loosely coupled staggered time-integration methods.

The paper by Hauke and Fuster presents a variational multiscale a posteriori error estimation method that is based on approximating an exact representation of the error based on fine-scale Green's function.

Kannan and Masud present two stabilized formulations for the Schrödinger wave equation. One is based on Galerkin/Least-squares ideas and the second one is based on the Variational Multiscale ideas. Using the generalized Kronig–Penney problem they present numerical convergence studies to demonstrate the accuracy and convergence properties of the two methods.

The papers presented in this special volume represent some of the recent advances in the stabilized and multiscale finite element methods and their application to a variety of problems. Lastly, we are very grateful to those who have contributed to the success of this special issue.

**Arif Masud**
University of Illinois at Urbana-Champaign

**Tayfun E. Tezduyar**
Rice University

**Yoichiro Matsumoto**
University of Tokyo

# Variational Multiscale A Posteriori Error Estimation for Quantities of Interest

**Guillermo Hauke**
e-mail: ghauke@unizar.es

**Daniel Fuster**

LITEC (CSIC) – Area de Mecanica de Fluidos,
Centro Politecnico Superior Zaragoza,
C/Maria de Luna 3,
50018 Zaragoza, Spain

*This paper applies the variational multiscale theory to develop an explicit a posteriori error estimator for quantities of interest and linear functionals of the solution. The method is an extension of a previous work on global and local error estimates for solutions computed with stabilized methods. The technique is based on approximating an exact representation of the error formulated as a function of the fine-scale Green function. Numerical examples for the multidimensional transport equation confirm that the method can provide good local error estimates of quantities of interest both in the diffusive and the advective limit.* [DOI: 10.1115/1.3057403]

## 1 Introduction

Lately much of attention of a posteriori error estimation has been placed in quantities of interest and functional outputs. Indeed, in engineering and science applications many times one is interested not just in the global solution, but in values or functions of the solution at particular places. In these cases, it can be argued that the computation should be driven by minimizing the error of these searched quantities (see Refs. [1,2] and references therein).

Previous work on error estimation for quantities of interest is typically based on the solution of two problems, the primal and the dual problem. The primal problem is the fundamental mathematical model that gives the desired solution, whereas the dual problem provides information on how the error in the primal problem influences the quantity of interest [1–4]. Applications of the adjoint technique to advection-diffusion problems can be found in Refs. [5,6] and references therein, and to methods involving stabilized methods in Refs. [7,8] and references therein.

This paper takes on a different point of view, namely, the variational multiscale theory [9,10], which avoids solving the dual problem. Furthermore, the method is set up as an explicit a posteriori error estimator, leading to a very economical technology.

The variational multiscale theory has been investigated for obtaining an estimate of the error distribution in elliptic problems in Ref. [11]. The technique to develop explicit a posteriori error estimators for the transport equation was presented in Ref. [12]. This approach has been shown exact for the class of edge-exact solutions [13–15] and has been extended to multidimensional transport problems in Ref. [16].

The success of the technique can be traced down to two reasons. The first one is that the method solves analytically a priori the dual problem. The second reason is that for the class of methods stemming from $H_0^1$ projection or optimization (like stabilized methods [17–19]), the error distribution is practically local [20]. This strategy is, therefore, well suited to advection dominated solutions computed with stabilized methods.

## 2 Preliminaries

This section reviews the point of departure.

**2.1 The Abstract Problem.** Consider a spatial domain $\Omega$ with boundary $\Gamma$. The strong form of the boundary value problem consists of finding $u:\Omega\rightarrow\mathbb{R}$ such that for the given essential boundary condition $g:\Gamma_g\rightarrow\mathbb{R}$, the natural boundary condition $h:\Gamma_h\rightarrow\mathbb{R}$, and forcing function $f:\Omega\rightarrow\mathbb{R}$, $f\in L_2$ (if $\Gamma_h=\emptyset$, $f\in H^{-1}$), the following equations are satisfied:

$$\begin{aligned} \mathcal{L}u &= f \quad \text{in } \Omega \\ u &= g \quad \text{on } \Gamma_g \\ \mathcal{B}u &= h \quad \text{on } \Gamma_h \end{aligned} \tag{1}$$

where $\mathcal{L}$ is in principle a second-order differential operator and $\mathcal{B}$ is an operator acting on the boundary, emanating from integration-by-parts of the weak form.

**2.2 The Variational Multiscale Error Estimation Paradigm.** The variational multiscale method [9] introduces a sum decomposition of the *exact solution* $u\in\mathcal{S}\subset H^1$ into the *finite element solution* (resolved scales) $\bar{u}$ and the *error* (unresolved or subgrid scales) $u'$

$$u=\bar{u}+u' \tag{2}$$

Typically $\bar{u}$ belongs to a finite element space $\bar{\mathcal{S}}$ with $\Omega^e$, $e=1,\ldots,n_{\text{el}}$ disjoint elements. The union of element interiors is denoted by $\tilde{\Omega}=\cup_e^{n_{\text{el}}}\Omega^e$, whereas the interelement boundaries are denoted by $\tilde{\Gamma}=\cup_e^{n_{\text{el}}}\Gamma^e\backslash\Gamma$, with $\Gamma^e$ as the element boundary. Accordingly, the error $u'\in\mathcal{S}'$ with $\mathcal{S}'=\mathcal{S}/\bar{\mathcal{S}}$.

Then, the error of the numerical computation can be calculated by the following paradigm [10,12]:

$$\begin{aligned} u'(\mathbf{x}) = &-\int_{\tilde{\Omega}_y} g'(\mathbf{x},\mathbf{y})(\mathcal{L}\bar{u}-f)(\mathbf{y})\mathrm{d}\Omega_y \\ &-\int_{\tilde{\Gamma}_y\cup\Gamma_{h_y}} g'(\mathbf{x},\mathbf{y})([\![\mathcal{B}\bar{u}]\!])(\mathbf{y})\mathrm{d}\Gamma_y \end{aligned} \tag{3}$$

where $\mathbf{x},\mathbf{y}\in\Omega$, $g'(\mathbf{x},\mathbf{y}):\Omega\times\Omega\rightarrow\mathbb{R}$ is the Green's function of the *fine-scale* problem [9,10,20], and $[\![\cdot]\!]$ is the generalization of the jump operator [10,21] to include the error on the natural boundary condition boundary

$$[\![\mathcal{B}\bar{u}]\!]\equiv\begin{cases} [\![\mathcal{B}\bar{u}]\!] & \text{on } \tilde{\Gamma} \\ \mathcal{B}\bar{u}-h & \text{on } \Gamma^e\cap\Gamma_h \\ 0 & \text{on } \Gamma^e\cap\Gamma_g \end{cases} \tag{4}$$

The fine-scale Green's function $g'(\mathbf{x},\mathbf{y})$ is the distribution that characterizes the behavior of the numerical error, and emanates from the proper projection of the global Green's function. There-

Contributed by the Applied Mechanics Division of ASME for publication in the JOURNAL OF APPLIED MECHANICS. Manuscript received October 31, 2007; final manuscript received July 8, 2008; published online January 13, 2009. Review conducted by Arif Masud.

fore, it depends on the operator (with the corresponding geometry and boundary conditions), on the finite element space, and on the method (or projector). Furthermore, according to Hughes and Sangalli [20] $\mathcal{S}'$ is the kernel of the projector that defines the method.

Following Hauke et al. [16], the error is split into contributions stemming from element interiors and interelement boundaries

$$u'(\mathbf{x}) = u'_{\text{int}}(\mathbf{x}) + u'_{\text{bnd}}(\mathbf{x}) \tag{5}$$

where

$$u'_{\text{int}}(\mathbf{x}) = -\int_{\tilde{\Omega}_y} g'(\mathbf{x},\mathbf{y})(\mathcal{L}\bar{u} - f)(\mathbf{y})\mathrm{d}\Omega_y$$

$$u'_{\text{bnd}}(\mathbf{x}) = -\int_{\tilde{\Gamma}_y \cup h_y} g'(\mathbf{x},\mathbf{y})([\![\mathcal{B}\bar{u}]\!])(\mathbf{y})\mathrm{d}\Gamma_y \tag{6}$$

## 3 Error of Quantities of Interest and Functional Outputs

**3.1 Functional Outputs.** We are interested in a quantity of interest $F(u)$, which is given by the linear functional of $u$

$$F(u) = \int_{\omega} K(\mathbf{x})u(\mathbf{x})\mathrm{d}\Omega \tag{7}$$

where $\omega$ is a spatial domain $\omega \subset \Omega$.

**3.2 Examples of Functionals.** Sections 3.2.1–3.2.3 show examples of functionals $F(u)$ and $K(\mathbf{x})$ to estimate the error at discrete points of the domain for element mean values and integrals of the solution on a boundary. For other particular cases, the expression of $F(u)$ and $K(\mathbf{x})$ can be deduced.

*3.2.1 Pointwise Error*. When it is required to control the error at a particular point, the following functions have to be used:

$$F(u) = u(\mathbf{x}_0) \tag{8}$$

$$K(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_0) \tag{9}$$

*3.2.2 Element Mean Value*. When it is required to control the mean element error, then

$$F(u) = \frac{1}{\Omega^e}\int_{\Omega^e} u(\mathbf{x})\mathrm{d}\Omega \tag{10}$$

$$K(\mathbf{x}) = \begin{cases} 1/\Omega^e, & \mathbf{x} \in \Omega^e \\ 0, & \text{the rest} \end{cases} \tag{11}$$

*3.2.3 Integral on a Boundary*. To control the error along the boundary of domain $\Gamma_{\text{out}}$

$$F(u) = \int_{\Gamma_{\text{out}}} u(\mathbf{x})\mathrm{d}\Gamma \tag{12}$$

$$K(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \Gamma_{\text{out}} \\ 0, & \text{the rest} \end{cases} \tag{13}$$

**3.3 Variational Multiscale Theory.** Since $F(u)$ is linear, given a finite element solution $\bar{u}$ the error in the quantity of interest can be calculated as

$$F(u') = \int_{\omega} K(\mathbf{x})u'(\mathbf{x})\mathrm{d}\Omega \tag{14}$$

Substituting the exact error representation (3)

$$F(u') = -\int_{\omega_x}\int_{\tilde{\Omega}_y} K(\mathbf{x})g'(\mathbf{x},\mathbf{y})(\mathcal{L}\bar{u} - f)(\mathbf{y})\mathrm{d}\Omega_x\mathrm{d}\Omega_y - \int_{\omega_x}\int_{\tilde{\Gamma}_y \cup \Gamma_{h_y}} K(\mathbf{x})g'(\mathbf{x},\mathbf{y})([\![\mathcal{B}\bar{u}]\!])(\mathbf{y})\mathrm{d}\Omega_x\mathrm{d}\Gamma_y \tag{15}$$

*3.3.1 Particular Case*. Assume that $\mathcal{L}\bar{u} - f \in \mathcal{P}_0$ (i.e., piecewise constant residual) and that $u' = 0$ on $\Gamma^e$. The domain $\omega$ is made of elements. Then, the error on $\tilde{\Gamma}_y \cup \Gamma_h$ vanishes, the fine-scale Green's function $g'(\mathbf{x},\mathbf{y})$ becomes the element Green's function $g_e(\mathbf{x},\mathbf{y})$ and

$$\begin{aligned} F(u') &= -\int_{\omega_x}\int_{\tilde{\Omega}_y} K(\mathbf{x})g'(\mathbf{x},\mathbf{y})(\mathcal{L}\bar{u} - f)(\mathbf{y})\mathrm{d}\Omega_x\mathrm{d}\Omega_y \\ &= -\sum_{\Omega^e \subset \omega} (\mathcal{L}\bar{u} - f)^e \int_{\Omega^e_x} K(\mathbf{x})\int_{\Omega^e_y} g_e(\mathbf{x},\mathbf{y})\mathrm{d}\Omega_x\mathrm{d}\Omega_y \\ &= -\sum_{\Omega^e \subset \omega} (\mathcal{L}\bar{u} - f)^e \int_{\Omega^e_x} K(\mathbf{x})b^e(\mathbf{x})\mathrm{d}\Omega_x \\ &= -\sum_{\Omega^e \subset \omega} (\mathcal{L}\bar{u} - f)^e F(b^e(\mathbf{x})) \end{aligned} \tag{16}$$

where $b^e(\mathbf{x})$ is the corresponding *residual-free bubble* [22–25] and $(\mathcal{L}\bar{u} - f)^e$ the residual in element $\Omega^e$. Therefore, the error on the functional can be calculated as the sum of elemental contributions. This result is a consequence of the linearity of $F(\cdot)$.

Recall that the assumption $u' = 0$ on $\Gamma^e$ decouples the error between elements, and the residual-free bubble $b^e(\mathbf{x})$ can be calculated elementwise.

**3.4 A Model for Error Estimation.** Since the previous expression is hard to compute analytically, for practical computations it is very convenient to introduce some approximations, similar to those in Ref. [16], which have been shown to be successful for a posteriori error estimation.

The error $F(u')$ is divided into two components,

$$F(u') = F_{\text{int}}(u') + F_{\text{bnd}}(u') \tag{17}$$

stemming, respectively, from the element interiors and the element interfaces,

$$\begin{aligned} F_{\text{int}}(u') = F(u'_{\text{int}}) &= \int_{\omega_x} K(\mathbf{x})u'_{\text{int}}(\mathbf{x})\mathrm{d}\Omega_x \\ &= -\int_{\omega_x}\int_{\tilde{\Omega}_y} K(\mathbf{x})g'(\mathbf{x},\mathbf{y}) \times (\mathcal{L}\bar{u} - f)(\mathbf{y})\mathrm{d}\Omega_x\mathrm{d}\Omega_y \end{aligned} \tag{18}$$

$$\begin{aligned} F_{\text{bnd}}(u') = F(u'_{\text{bnd}}) &= \int_{\omega_x} K(\mathbf{x})u'_{\text{bnd}}(\mathbf{x})\mathrm{d}\Omega_x \\ &= -\int_{\omega_x}\int_{\tilde{\Gamma}_y \cup \Gamma_{h_y}} K(\mathbf{x})g'(\mathbf{x},\mathbf{y}) \times ([\![\mathcal{B}\bar{u}]\!])(\mathbf{y})\mathrm{d}\Omega_x\mathrm{d}\Gamma_y \end{aligned} \tag{19}$$

*Remark*. The error emanating from element interior residuals is the main contribution of the error in the advection dominated regime, whereas the interelement boundary term is crucial for predicting the error in the diffusion dominated regime [16].

Applying twice Hölder's inequality to $F_{\text{int}}(u')$ (see Ref. [26])

$$|F_{\text{int}}(u')| \leq \|K(\mathbf{x})\|_{L_{p'}(\omega)} \|\|g'(\mathbf{x},\mathbf{y})\|_{L_p(\Omega_y)} \|_{L_{q'}(\omega_x^e)} \|\mathcal{L}\bar{u} - f\|_{L_q(\Omega)} \tag{20}$$

with $1 \leq p,q \leq \infty$, $1 \leq p',q' \leq \infty$, $1/p+1/q=1$, and $1/p'+1/q'=1$. Likewise, for the interelement boundary error

$$|F_{\text{bnd}}(u')| \leq \|K(\mathbf{x})\|_{L_{p'}(\omega)} \|\|g'(\mathbf{x},\mathbf{y})\|_{L_p(\tilde{\Gamma}_y \cup \Gamma_{h_y})} \|_{L_{q'}(\omega_x^e)} \|[\![\mathcal{B}\bar{u}]\!]\|_{L_q(\Omega)} \tag{21}$$

with the same conditions on $p$, $q$, $p'$, and $q'$.

*3.4.1 Element Interior Error.* The computation of the exact error requires full knowledge of the fine-scale Green's function $g'(\mathbf{x},\mathbf{y})$, which can be analytically or computationally involved. However, for the class of variational methods, such as stabilized methods, where the error distribution is practically local [20], the fine-scale Green's function can be approximated by the element Green's function $g_e(x,y)$, which for linear elements satisfies within each element

$$\mathcal{L}g_e = \delta_{\mathbf{y}} \quad \text{in } \Omega^e$$

$$g_e = 0 \quad \text{on } \Gamma^e \tag{22}$$

where $\delta_{\mathbf{y}}(\mathbf{x}) = \delta(\mathbf{x}-\mathbf{y})$ represents the Dirac delta distribution.

Following Hauke et al. [12–14] the error due to element interiors is modeled as

$$\begin{aligned} u'_{\text{int}}(\mathbf{x}) &= -\int_{\tilde{\Omega}_y} g'(\mathbf{x},\mathbf{y})(\mathcal{L}\bar{u} - f)(\mathbf{y})\mathrm{d}\Omega_y \\ &\approx -\int_{\Omega_y^e} g_e(\mathbf{x},\mathbf{y})(\mathcal{L}\bar{u} - f)(\mathbf{y})\mathrm{d}\Omega_y \quad \text{on } \Omega^e \end{aligned} \tag{23}$$

The preceding paradigm (23) is exact for element-edge-exact solutions. This is the case of one-dimensional linear problems solved with stabilized methods or one-dimensional Poisson problems solved with the Galerkin method.

Substituting in Eq. (18) and assuming that $\omega$ is the union of various elements

$$\begin{aligned} F_{\text{int}}(u') &= \int_{\omega_x} K(\mathbf{x}) u'_{\text{int}}(\mathbf{x})\mathrm{d}\Omega_x \\ &= \sum_{\Omega^e \subset \omega} \int_{\Omega^e} K(\mathbf{x}) u'_{\text{int}}(\mathbf{x})\mathrm{d}\Omega_x \\ &\approx -\sum_{\Omega^e \subset \omega} \int_{\Omega_x^e} \int_{\Omega_y^e} K(\mathbf{x}) g_e(\mathbf{x},\mathbf{y}) \\ &\quad \times (\mathcal{L}\bar{u} - f)(\mathbf{y})\mathrm{d}\Omega_x \mathrm{d}\Omega_y \end{aligned} \tag{24}$$

Applying Hölder's inequality twice

$$\begin{aligned} |F_{\text{int}}(u')| \leq & \sum_{\Omega^e \subset \omega} \|K(\mathbf{x})\|_{L_{p'}(\Omega^e)} \|\|g_e(\mathbf{x},\mathbf{y})\|_{L_p(\Omega_y^e)}\|_{L_{q'}(\Omega_x^e)} \\ & \times \|\mathcal{L}\bar{u} - f\|_{L_q(\Omega^e)} \end{aligned} \tag{25}$$

with $1 \leq p,q \leq \infty$, $1 \leq p',q' \leq \infty$, $1/p+1/q=1$, and $1/p'+1/q'=1$. Now the norm of the element Green's function can be written as a function of the error time scales, which depend on the choice of the above parameters.

*3.4.2 Element Boundary Error.* The interelement boundary errors are approximated within each element as

$$\begin{aligned} u'_{\text{bnd}}(\mathbf{x}) &= -\int_{\Gamma_y \cup \Gamma_{h_y}} g'(\mathbf{x},\mathbf{y})([\![\mathcal{B}\bar{u}]\!])(\mathbf{y})\mathrm{d}\Gamma_y \\ &\approx -\int_{\Gamma_y^e} g'(\mathbf{x},\mathbf{y})([\![\mathcal{B}\bar{u}]\!])(\mathbf{y})\mathrm{d}\Gamma_y \quad \text{on } \Omega^e \end{aligned} \tag{26}$$

Substituting in Eq. (19) and assuming that $\omega$ is the union of various elements

$$\begin{aligned} F_{\text{bnd}}(u') &= \int_{\omega_x} K(\mathbf{x}) u'_{\text{bnd}}(\mathbf{x})\mathrm{d}\Omega_x \\ &= \sum_{\Omega^e \subset \omega} \int_{\Omega_x^e} K(\mathbf{x}) u'_{\text{bnd}}(\mathbf{x})\mathrm{d}\Omega_x \\ &\approx -\sum_{\Omega^e \subset \omega} \int_{\Omega_x^e} K(\mathbf{x}) \int_{\Gamma_y^e} g'(\mathbf{x},\mathbf{y}) \\ &\quad \times ([\![\mathcal{B}\bar{u}]\!])(\mathbf{y})\mathrm{d}\Omega_x \mathrm{d}\Gamma_y \end{aligned} \tag{27}$$

Again, by Hölder's inequality [13]

$$|F_{\text{bnd}}(u')| \leq \sum_{\Omega^e \subset \omega} \|K(\mathbf{x})\|_{L_{p'}(\Omega^e)} \|\|g'(\mathbf{x},\mathbf{y})\|_{L_p(\Gamma_y^e)}\|_{L_{q'}(\Omega_x^e)} \|[\![\mathcal{B}\bar{u}]\!]\|_{L_q(\Gamma^e)} \tag{28}$$

with $1 \leq p,q \leq \infty$, $1 \leq p',q' \leq \infty$, $1/p+1/q=1$, and $1/p'+1/q'=1$.

## 4 Error Estimation Paradigm

**4.1 Selection of Norms.** The proposed method for error estimation requires choosing the parameters $p$, $q$, $p'$, and $q'$ with the conditions of Secs. 3.4.1 and 3.4.2.

Generally speaking, the element Green's function can be complicated to calculate. Selecting $p=1$ can ease this task because if the element Green's function does not change sign within the element, its $L_1$ norm equals the corresponding residual-free bubble. Residual-free bubbles are smoother functions and are much simpler to calculate than Green's functions. On the other hand, the kernel $K(x)$ can also be a distribution or a rough function, so choosing $p'=1$ is also a wise selection. Sometimes, such as when the kernel is a Dirac delta, that may even be the only possible choice.

Therefore, for the numerical examples shown in Sec. 4.2, the values of these parameters are

| $p$ | $q$ | $p'$ | $q'$ |
|---|---|---|---|
| 1 | $\infty$ | 1 | $\infty$ |

**4.2 The Norm of the Fine-Scale Green's Function at the Boundary.** In Ref. [16] it was shown that the norm of the fine-scale Green's function $g'(x,y)$ on $\Gamma^e$ can be approximated to the norm of the element Green's function in the domain $\Omega^e$ by

$$\|\|g'(\mathbf{x},\mathbf{y})\|_{L_1(\Gamma_y^e)}\|_{L_r(\Omega_x^e)} \approx \frac{1}{2}\frac{\text{meas}(\Gamma^e)}{\text{meas}(\Omega^e)} \|\|g_e(\mathbf{x},\mathbf{y})\|_{L_1(\Omega_y^e)}\|_{L_r(\Omega_x^e)} \tag{29}$$

The factor $\frac{1}{2}$ appears due to the splitting of the boundary residual error, which has to be considered as a first approximation.

**4.3 Error Estimation Representation.** Once these parameters are selected, an upper bound of the measure of the error can be estimated applying Eq. (28) and using Eq. (29)

$$
\begin{aligned}
|F(u')| &= |F_{\text{int}}(u') + F_{\text{bnd}}(u')| \\
&< |F_{\text{int}}(u')| + |F_{\text{bnd}}(u')| \\
&\leq \sum_{\Omega^e \subset \omega} \|K(\mathbf{x})\|_{L_1(\Omega^e)} \tau_{1\infty}^e \left( \|\mathcal{L}\bar{u} - f\|_{L_\infty(\Omega^e)} \right. \\
&\quad \left. + \frac{1}{2} \frac{\text{meas}(\Gamma^e)}{\text{meas}(\Omega^e)} \|[\![\mathcal{B}\bar{u}]\!]\|_{L_\infty(\Gamma^e)} \right)
\end{aligned} \tag{30}
$$

where $\tau_{1\infty}^e$ is an approximation of the norm of the element Green's function $\|\|g_e(\mathbf{x},\mathbf{y})\|_{L_1(\Omega_y^e)}\|_{L_\infty(\Omega_x^e)}$ calculated from one-dimensional functions [14,16]. In particular, for the advection-diffusion-reaction equation employed in Sec. 5,

$$
\tau_{1\infty}^e = \min\left( \frac{h_{\text{flow}}^e}{|\mathbf{a}|}; \frac{(h_x^e)^2}{8\kappa}; \frac{1}{|s|} \right) \tag{31}
$$

being $h_x^e$ the element side length and $h_{\text{flow}}^e$ the element length in the streamline direction. Hauke et al. [16] showed that the error involved in this approximation is small.

## 5 Numerical Examples

In order to validate the proposed method, the error estimator is applied to solutions of the transport equation

$$
\begin{gathered}
\mathcal{L}u \equiv \mathbf{a} \cdot \nabla u - \nabla \cdot (\kappa \nabla u) - su = 0 \quad \text{in } \Omega \\
u = \text{g} \quad \text{on } \Gamma_g \\
\mathcal{B}u \equiv \kappa \frac{\partial u}{\partial n} = \text{h} \quad \text{on } \Gamma_h
\end{gathered} \tag{32}
$$

where $\mathbf{a}$ is the velocity field, $\kappa \geq 0$ the diffusion coefficient, and $s$ the source parameter.

The quantity of interest is taken as the pointwise error (8) where the kernel is the Dirac delta distribution. Two cases are addressed: element nodes and element centers.

(a) *Element nodes*. The kernel is considered to be spread over the four elements around the node. For the structured meshes of quads considered in this section, for each element that includes the node

$$
\|K(\mathbf{x})\|_{L_1(\Omega^e)} = \int_{\Omega^e} |K(\mathbf{x})| \text{d}\Omega = \frac{1}{4} \tag{33}
$$

(b) *Element centers*. The kernel is considered spread over the entire element. In this case, only one element enters into the error estimate

$$
\|K(\mathbf{x})\|_{L_1(\Omega^e)} = \int_{\Omega^e} |K(\mathbf{x})| \text{d}\Omega = 1 \tag{34}
$$

These two tests give a global view of the error estimator efficiency around the domain. Other quantities of interest (e.g., the integration of the transported variable along a particular area) will be closely related to the pointwise error distribution around the domain.

It is also important to remark that the information required to compute the error is minimal. In this case it is restricted to the elements that share the point where the error is estimated. Therefore, the computational cost of the present error estimator is very light.

The numerical solution is calculated using a stabilized method [17–19]. Throughout the study, the flow time scale $\tau_{\text{flow}}^e$ of the stabilized method is evaluated as the modified Franca–Valentin tau, $\tau_{\text{mfv}}^e$, whose definition can be found in Refs. [27,28]. Also, the streamwise element length is calculated as $h_{\text{flow}}^e = h_x^e / \cos\alpha$, where $\alpha$ is the flow angle ($\alpha = \arctan u_y/u_x$).

The error estimator of the pointwise error is applied to the example of the square domain problem described in Ref. [29]. The independent force term is calculated such that the exact solution is

$$
\begin{aligned}
u(x,y) &= xy^2 - y^2 \exp\left( \frac{2(x-1)}{\kappa} \right) - x \exp\left( \frac{3(y-1)}{\kappa} \right) \\
&\quad + \exp\left( \frac{2(x-1) + 3(y-1)}{\kappa} \right)
\end{aligned} \tag{35}
$$

where the parameters are $\mathbf{a} = (2,3)$, $s = -1$, and for this case, two values of $\kappa = 10^{-6}, 1$ are tested.

It must be noticed that due to the synthetic nature of the solution, it is characterized by nonhomogeneous boundary conditions, which must be calculated using Eq. (35). These values have been imposed at the boundary nodes.

The efficiency of the absolute value of the pointwise error is obtained at the mesh nodes and element centers. The results for both estimations are plotted in Fig. 1 for the two considered viscosities. The efficiency has been plotted as a function of the real error $F(u')$ in a logarithmic scale. As can be seen, the efficiencies are close to 1 at those points where the solution is more abrupt and the committed error is larger. This trend is encountered for all the tests, where it is also verified that, in general, when the error is large enough, the estimation provided by the current method is an upper bound of the real error.

Figures 2 and 3 depict an interpolation of the local efficiencies shown by the contour levels over the representation of the spatial exact error distribution.

For the advective solution (Fig. 2) the error is concentrated around the sharp gradients, which are generated at the outflow boundary layers, for $y$ tending to 1. The more refined the mesh is, the less the error, although the efficiencies in those zones where the error is large remain in values under 5.

For the diffusive solution (Fig. 3), the error is smaller due to the smooth nature of the solution. The error is more homogeneously distributed around the domain and again, except for those zones where the error becomes almost null, the efficiencies take values of between 1 and 10.

## 6 Conclusions

An explicit a posteriori error estimator for quantities of interest based on linear functionals has been developed from the variational multiscale theory. The technique includes norms of both interior element residuals and interelement residuals. The error time scales, which also represent error constants, have been obtained explicitly from element Green's functions.

The efficiency of the method for estimating the pointwise error at mesh nodes and element centers has been tested in advection-diffusion-reaction problems, including advection dominated and diffusion dominated flows. In all cases, the global efficiencies are close to 1 when the errors are large. Furthermore, the numerical experiments have also shown that for all the ranges of considered parameters, the local efficiencies are a good approximation of the true error, mainly in areas where the errors are large.

Thus, the proposed a posteriori error estimator leads to a very economical and robust technique for transport problems computed with stabilized methods.


## Acknowledgment

This work has been partially funded by the Ministerio de Ciencia y Tecnología under Contract No. MCYT VEM2003-20069-C03-01.

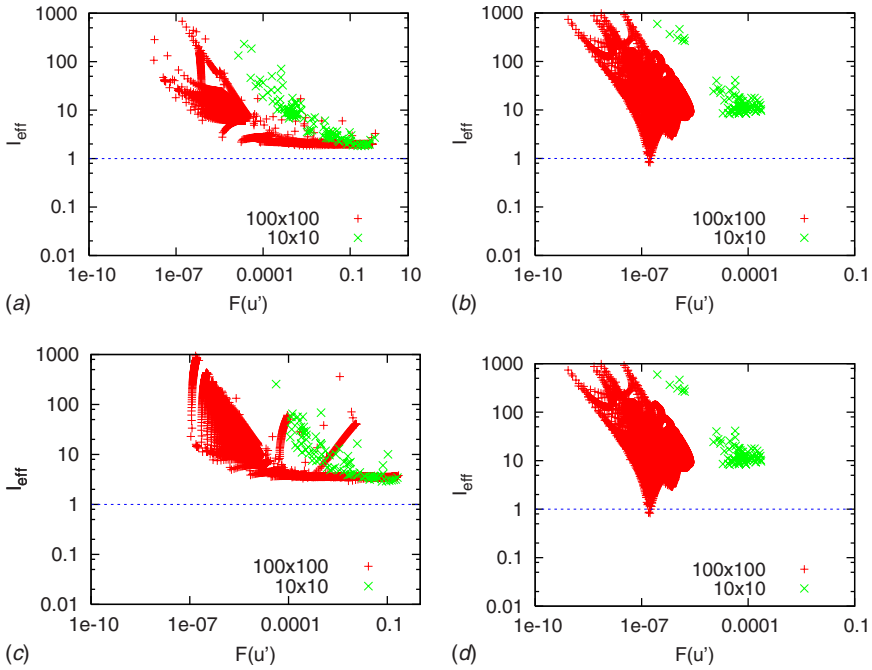



**Fig. 1 Local efficiencies for two different meshes (10×10 and 100×100) for a diffusive problem (right) and an advective problem (left). Top graphs: error at the element centers. Bottom graphs: error at the mesh nodes.**

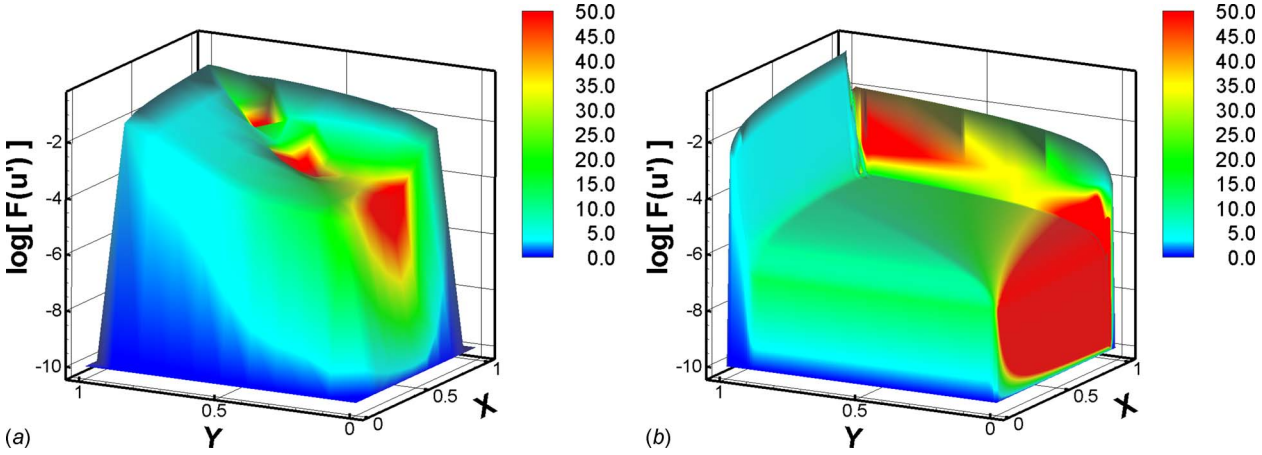



**Fig. 2 Advection dominated problem. Efficiency of the pointwise error at the mesh nodes. Meshes of 10×10 (left) and 100×100 (right).**

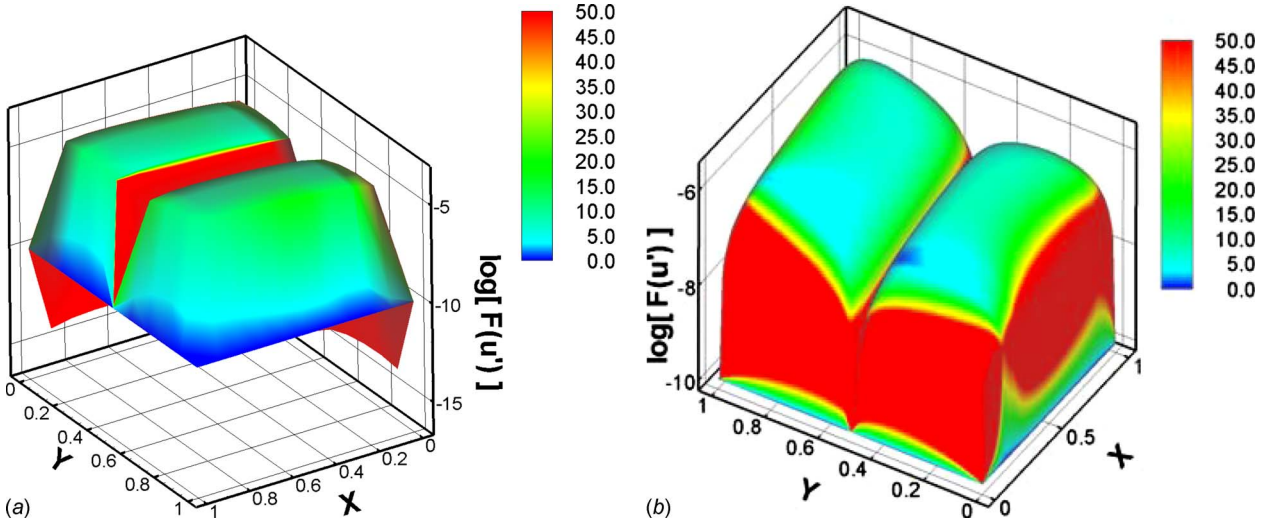



**Fig. 3 Diffusion dominated problem. Efficiency of the pointwise error at the mesh nodes. Meshes of 10×10 (left) and 100×100 (right).**

## References


[1] Ainsworth, M., and Oden, J. T., 2000, *A Posterior Error Estimation in Finite Element Analysis*, Wiley, New York.

[2] Bangerth, W., and Rannacher, R., 2003, *Adaptive Finite Element Methods for Differential Equations*, Birkhäuser, Basel.

[3] Pares, N., Diez, P., and Huerta, A., 2006, "Subdomain-Based Flux-Free A Posteriori Error Estimators," Comput. Methods Appl. Mech. Eng., **195**, pp. 297–323.

[4] Prudhomme, S., and Oden, J. T., 1999, "On Goal-Oriented Error Estimation for Elliptic Problems: Application to the Control of Pointwise Errors," Comput. Methods Appl. Mech. Eng., **176**, pp. 313–331.

[5] Paraschivoiu, M., Peraire, J., and Patera, A. T., 1997, "A Posteriori Finite Element Bounds for Linear-Functional Outputs of Elliptic Partial Differential Equations," Comput. Methods Appl. Mech. Eng., **150**, pp. 289–312.

[6] Peraire, J., and Patera, A. T., 1999, "Asymptotic A Posteriori Finite Element Bounds for the Outputs of Noncoercive Problems: The Helmholtz and Burgers Equations," Comput. Methods Appl. Mech. Eng., **171**, pp. 77–86.

[7] Houston, P., Rannacher, R., and Süli, E., 2000, "A Posteriori Error Analysis for Stabilized Finite Element Approximations of Transport Problem," Comput. Methods Appl. Mech. Eng., **190**, pp. 1483–1508.

[8] Rannacher, R., 1998, "A Posteriori Error Estimation in Least-Squares Stabilized Finite Element Schemes," Comput. Methods Appl. Mech. Eng., **166**, pp. 99–114.

[9] Hughes, T. J. R., 1995, "Multiscale Phenomena: Green's Functions, the Dirichlet-to-Neumann Formulation, Subgrid Scale Models, Bubbles and the Origins of Stabilized Methods," Comput. Methods Appl. Mech. Eng., **127**, pp. 387–401.

[10] Hughes, T. J. R., Feijoo, G. R., Mazzei, L., and Quincy, J. B., 1998, "The Variational Multiscale Method: A Paradigm for Computational Mechanics," Comput. Methods Appl. Mech. Eng., **166**, pp. 3–24.

[11] Larson, M. G., and Målqvist, A., 2007, "Adaptive Variational Multiscale Methods Based on A Posteriori Error Estimation: Energy Norm Estimates for Elliptic Problems," Comput. Methods Appl. Mech. Eng., **196**, pp. 2313–2324.

[12] Hauke, G., Doweidar, M. H., and Miana, M., 2006, "The Multiscale Approach to Error Estimation and Adaptivity," Comput. Methods Appl. Mech. Eng., **195**, pp. 1573–1593.

[13] Hauke, G., and Doweidar, M. H., 2006, "Intrinsic Scales and A Posteriori Multiscale Error Estimation for Piecewise-Linear Functions and Residuals," Int. J. Comput. Fluid Dyn., **20**, pp. 211–222.

[14] Hauke, G., Doweidar, M. H., and Miana, M., 2006, "Proper Intrinsic Scales for A-Posteriori Multiscale Error Estimation," Comput. Methods Appl. Mech. Eng., **195**, pp. 3983–4001.

[15] Hauke, G., Doweidar, M. H., Fuster, D., Gomez, A., and Sayas, J., 2006, "Application of Variational A-Posteriori Multiscale Error Estimation to Higher-Order Elements," Comput. Mech., **38**, pp. 382–389.

[16] Hauke, G., Fuster, D., and Doweidar, M. H., 2008, "Variational Multiscale A-Posteriori Error Estimation for Multi-Dimensional Transport Problems," Comput. Methods Appl. Mech. Eng., **197**, pp. 2701–2718.

[17] Brooks, A. N., and Hughes, T. J. R., 1982, "Streamline Upwind/Petrov-Galerkin Formulations for Convection Dominated Flows With Particular Emphasis on the Incompressible Navier-Stokes Equations," Comput. Methods Appl. Mech. Eng., **32**, pp. 199–259.

[18] Franca, L. P., Frey, S. L., and Hughes, T. J. R., 1992, "Stabilized Finite Element Methods: I. Application to the Advective-Diffusive Model," Comput. Methods Appl. Mech. Eng., **95**, pp. 253–276.

[19] Franca, L. P., Hauke, G., and Masud, A., 2006, "Revisiting Stabilized Finite Element Methods for the Advective-Diffusive Equation," Comput. Methods Appl. Mech. Eng., **195**, pp. 1560–1572.

[20] Hughes, T. J. R., and Sangalli, G., 2007, "Variational Multiscale Analysis: The Fine-Scale Green's Function, Projection, Optimization, Localization and Stabilized Methods," SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal., **45**(2), pp. 539–557.

[21] Hughes, T. J. R., 2000, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Dover, New York.

[22] Brezzi, F., Bristeau, M. O., Franca, L. P., Mallet, M., and Rogé, G., 1992, "A Relationship Between Stabilized Finite Element Methods and the Galerkin Method With Bubble Functions," Comput. Methods Appl. Mech. Eng., **96**, pp. 117–129.

[23] Brezzi, F., Franca, L. P., Hughes, T. J. R., and Russo, A., 1997, "$b=\int g$," Comput. Methods Appl. Mech. Eng., **145**, pp. 329–339.

[24] Brezzi, F., and Russo, A., 1994, "Choosing Bubbles for Advection-Diffusion Problems," Math. Models Meth. Appl. Sci., **4**, pp. 571–587.

[25] Franca, L. P., and Russo, A., 1996, "Deriving Upwinding, Mass Lumping and Selective Reduced Integration by Residual-Free Bubbles," Appl. Math. Lett., **9**, pp. 83–88.

[26] Brenner, S. C., and Scott, L. R., 2002, *The Mathematical Theory of Finite Element Methods*, Springer-Verlag, New York.

[27] Franca, L. P., and Valentin, F., 2000, "On an Improved Unusual Stabilized Finite Element Method for the Advective-Reactive-Diffusive Equation," Comput. Methods Appl. Mech. Eng., **190**, pp. 1785–1800.

[28] Hauke, G., 2002, "A Simple Stabilized Method for the Advection-Diffusion-Reaction Equation," Comput. Methods Appl. Mech. Eng., **191**, pp. 2925–2947.

[29] John, V., 2000, "A Numerical Study of A Posteriori Error Estimators for Convection-Diffusion Equations," Comput. Methods Appl. Mech. Eng., **190**, pp. 757–781.

# Computational Modeling of the Collapse of a Liquid Column Over an Obstacle and Experimental Validation

**Marcela A. Cruchaga**
Departamento de Ingeniería Mecánica,
Universidad de Santiago de Chile (USACH),
Avenida Libertador Bernardo O´Higgins 3363,
Santiago, Chile
e-mail: mcruchag@lauca.usach.cl

**Diego J. Celentano**
Departamento de Ingeniería Mecánica y Metalúrgica,
Pontificia Universidad Católica de Chile,
Avenida Vicuña Mackenna 4860,
Santiago, Chile

**Tayfun E. Tezduyar**
Team for Advanced Flow Simulation and Modeling (T*AFSM),
Mechanical Engineering,
Rice University,
MS 321,
Houston, TX 77005

*We present the numerical and experimental analyses of the collapse of a water column over an obstacle. The physical model consists of a water column initially confined by a closed gate inside a glass box. An obstacle is placed between the gate and the right wall of the box, inside the initially unfilled zone. Once the gate is opened, the liquid spreads in the container and over the obstacle. Measurements of the liquid height along the walls and a middle control section are obtained from videos. The computational modeling is carried out using a moving interface technique, namely, the edge-tracked interface locator technique, to calculate the evolution of the water-air interface. The analysis involves a water-column aspect ratio of 2, with different obstacle geometries. The numerical predictions agree reasonably well with the experimental trends.*
[DOI: 10.1115/1.3057439]

*Keywords: moving interfaces, two-fluid flows, computational fluid mechanics, ETILT, experimental validation*

## 1 Introduction

Free-surface flow experiments at laboratory scales are commonly used for representing real problems, and they are useful in assessing the performance of the modeling by comparing the numerical predictions with the measured data. Several numerical techniques capable of accurately representing the evolution of a two-fluid interface can be found in the literature (e.g., see Refs. [1–16] and references therein). Focusing on the collapse of a water column, experiments were presented in Refs. [3,17]. This problem was adopted as a benchmark test to validate the numerical performance of the proposed free-surface flow formulations (e.g., see Refs. [3,7,10–13,15]). In particular, we presented in Ref. [15] a set of experiments and their corresponding simulation using a fixed-mesh finite element technique. The interface is "captured" with the edge-tracked interface locator technique (ETILT), introduced in Ref. [4], using the version described in Ref. [14].

In this paper, we report results from experiments for the collapse of a water column with an aspect ratio (height to width ratio of the initial liquid column) of 2 over obstacles with different geometries. The experimental data are used for evaluating the performance of the numerical strategy presented in Ref. [14] to model this problem. In particular, the computational parameters involved in the formulation, originally determined by numerical trial for the collapse of water column without an obstacle, are used in the present analysis to evaluate their independence from the geometry. Hence, in the present work, we test the performance of such parameters under different conditions. The dissipative interface capturing technique, used for inhibiting the formation of unrealistic bubbles in the fluid bulk, is now tested when physical gaps are formed, e.g., downstream of the obstacles. Moreover, the dissipative effect of the turbulence model is also assessed.

The governing equations are presented in Sec. 2. The ETILT and the new aspects included in its current version are summarized in Sec. 3. The details of the experimental procedure are described in Sec. 4. Experimental and numerical results are presented and discussed in Sec. 5. Concluding remarks are given in Sec. 6.

## 2 Governing Equations

The Navier–Stokes equations of unsteady incompressible flows are written as follows:

$$\rho\frac{\partial \mathbf{u}}{\partial t} + \rho\mathbf{u}\nabla\cdot\mathbf{u} + \nabla p - \nabla\cdot(2\mu\boldsymbol{\varepsilon}) = \rho\mathbf{f} \quad \text{in} \quad \Omega\times Y \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0 \quad \text{in} \quad \Omega\times Y \tag{2}$$

where $\rho$, $\mathbf{u}$, $p$, $\mu$, $\boldsymbol{\varepsilon}$, and $\mathbf{f}$ are the density, velocity, pressure, dynamic viscosity, strain-rate tensor, and the specific body force. In these equations, $\Omega$ denotes an open-bounded domain with a smooth boundary $\Gamma$, and $Y$ is the time interval of interest. This system of equations is completed with a set of initial and boundary conditions

$$\mathbf{u} = \mathbf{u}_0 \quad \text{in} \quad \Omega \tag{3}$$

$$\mathbf{u} = \mathbf{g} \quad \text{in} \quad \Gamma_g\times Y \tag{4}$$

$$\boldsymbol{\sigma}\cdot\mathbf{n} = \boldsymbol{h} \quad \text{in} \quad \Gamma_h\times Y \tag{5}$$

where $\mathbf{u}_0$ is the initial value of the velocity field, $\mathbf{g}$ represents the velocity imposed on the part of the boundary $\Gamma_g$, and $\boldsymbol{h}$ is the traction vector imposed over $\Gamma_h$ ($\Gamma_g\cup\Gamma_h=\Gamma$ and $\Gamma_g\cap\Gamma_h=\emptyset$), typically taken as a traction-free condition: $\boldsymbol{h}=0$.

In the present simulation, a simple model to compute the energy dissipated by turbulent effects can be considered by replacing $\mu$ in Eq. (1) with $\mu_t$ defined as

$$\mu_t = \min(\mu + l_{\text{mix}}^2\rho\sqrt{2\boldsymbol{\varepsilon}:\boldsymbol{\varepsilon}};\mu_{\max}) \tag{6}$$

where $l_{\text{mix}}$ is a characteristic mixing length. In the present work, $l_{\text{mix}}=C_t h_{UGN}$ with $C_t$ being a modeling parameter, $h_{UGN}$ a characteristic element length (see Ref. [8]), and $\mu_{\max}$ is a cut-off value.

Contributed by the Applied Mechanics Division of ASME for publication in the JOURNAL OF APPLIED MECHANICS. Manuscript received November 5, 2007; final manuscript received July 18, 2008; published online January 13, 2009. Review conducted by Arif Masud.

## 3 Interface Update

The interface between the two fluids (Fluid 1 and Fluid 2) represents a strong discontinuity in the fluid properties and the gradients of the velocity and pressure. Nevertheless, these variables are interpolated as continuous functions across the interface. Other types of discontinuities at the interface, e.g., surface tension, are not included in the present model. The interface motion is governed by an advection equation

$$\frac{\partial \varphi}{\partial t} + \mathbf{u} \cdot \nabla \varphi = 0 \quad \text{in} \quad \Omega \times Y \tag{7}$$

where $\varphi$ is a function marking the location of the interface. In the context of two-fluid flows, all the matrices and vectors associated with the finite element formulation of Eqs. (1) and (2) are computed while taking into account the discontinuities in the fluid properties.

In the present work, the interface is updated with the ETILT, introduced in Ref. [4], using the version described in Ref. [14]. From here onwards we summarize the technique following the referred work and references therein. At each time step, the density and viscosity distributions are obtained from

$$\rho^h = \varphi^{he}\rho_1 + (1 - \varphi^{he})\rho_2 \tag{8}$$

$$\mu^h = \varphi^{he}\mu_1 + (1 - \varphi^{he})\mu_2 \tag{9}$$

where $\varphi^{he}$ is the edge-based representation of $\varphi$. To compute $\varphi^{he}_{n+1}$ at time level $n+1$, given $\varphi^{he}_n$ at time level $n$, first a nodal representation $\varphi^h$ is computed. This is done by using a constrained least-squares projection as given in [9,15,16]

$$\int_\Omega \psi^h(\varphi^h_n - \varphi^{he}_n)d\Omega + \sum_{k=1}^{n_{ie}} \psi^h(\mathbf{x}_k)\lambda_{\text{pen}}(\varphi^h_n(\mathbf{x}_k) - 0.5) = 0 \tag{10}$$

Here $\psi^h$ is the test function, $n_{ie}$ is the number of the interface edges (i.e., the edges crossed by the interface regardless the problem dimension), $\mathbf{x}_k$ is the coordinate of the interface location along the $k$th interface edge, and $\lambda_{\text{pen}}$ is a penalty parameter. After this projection, we update the interface by using a discrete formulation of Eq. (7)

$$\int_\Omega \psi^h\left(\frac{\partial \varphi^h}{\partial t} + \mathbf{u}^h \cdot \nabla \varphi^h\right)d\Omega + \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} (\tau_{\text{SUPG}}\mathbf{u}^h \cdot \nabla \psi^h) \times \left(\frac{\partial \varphi^h}{\partial t} + \mathbf{u}^h \cdot \nabla \varphi^h\right)d\Omega + \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \nabla \psi^h \nu_{\text{DCID}} \nabla \varphi^h d\Omega = 0 \tag{11}$$

Here $n_{\text{el}}$ is the number of elements, $\tau_{\text{SUPG}}$ is the streamline upwind Petrov–Galerkin (SUPG) stabilization parameter [8], and $\nu_{\text{DCID}}$ is the discontinuity-capturing interface dissipation (DCID) parameter

$$\nu_{\text{DCID}} = \frac{C_s}{2}h^2_{UGN}\sqrt{2\boldsymbol{\varepsilon}:\boldsymbol{\varepsilon}}\frac{\|\nabla \varphi^h\|h_{UGN}}{\varphi_{\text{ref}}} \tag{12}$$

Here $C_s$ is a discontinuity-capturing constant and $\varphi_{\text{ref}}$ is a reference value, here set to 1. We note that the justification behind the expression given by Eq. (12) is combining some of the features we see in Eq. (7) and the discontinuity-capturing directional dissipation (DCDD) given in Ref. [8]. In time integration, $\varphi^h_{n+1}$ is computed from Eq. (11) by using the Crank–Nicholson scheme.

From $\varphi^h_{n+1}$ we obtain $\varphi^{he}_{n+1}$ by a combination of a least-squares projection and corrections to enforce volume conservation for chunks of Fluid 1 and Fluid 2 (see Ref. [15] for more details).

## 4 Experimental Procedure

In the present work, a simple experimental setup was built to obtain experimental data that the numerical results to be presented in Sec. 5 can be compared with. The apparatus, shown in Fig. 1, consists of a glass box with its two parts delimited by a gate with a mechanical release system. The left part of the box is initially filled with colored water. Only experiments with a column aspect ratio of $A_r=2$ ($A_r=H/L$, with $H$ and $L$ being the initial height and width of the column, respectively, with $L=0.114$ m) are presented. An obstacle is placed at $3/2L$ from the gate. The obstacles with different geometries are also shown in Fig. 1.

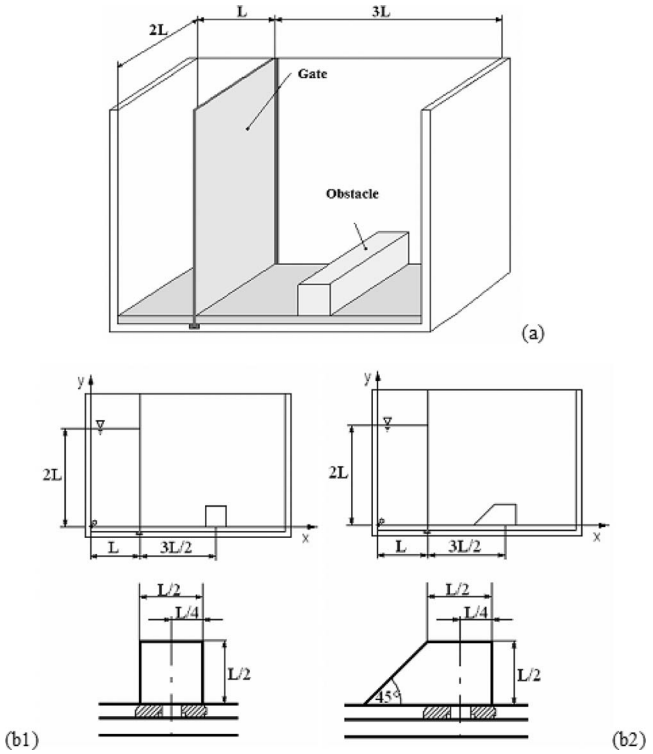



**Fig. 1 Schematic representation of the physical models (*a*) and geometry of obstacles ((*b*)(1) square section and (*b*)(2) trapezoidal section)**

Different experiments were carried out and videotaped. To measure the interface position, a ruler with tics 0.02 m apart is drawn surrounding the box and over the frame of the gate. A red marker that jointly moves with the gate helps to observe its position during the gate opening. The measurements reported in the present work are average values from the measurements registered for a minimum of five experiments. The maximum experimental errors in space and time are estimated as $\pm 0.005$ m and $\pm 0.025$ s, respectively.

The observed evolutions of the interface at the left and right walls of the box and in a vertical section in the middle of the obstacle are presented in Sec. 5 together with the corresponding numerical predictions.

## 5 Comparison of Numerical and Experimental Results for $A_r=2$

In this paper, the numerical simulations are focused on the long-term transient behavior for the collapse of the water column described in Sec. 4. In the simulations, the liquid column is initially at rest and confined between the left wall and the gate. The pressure is set to zero at the top of the rectangular computational domain. Slip conditions are assumed at the solid surfaces. The mesh is composed of $100 \times 75$ four-noded elements. The time-step size is 0.001 s. The fluid properties are $\rho_1=1000$ kg/m$^3$ and $\mu_1=0.001$ kg/m s$^{-1}$ for the water, and $\rho_2=1$ kg/m$^3$ and $\mu_2=0.001$ kg/m s$^{-1}$ for the air.

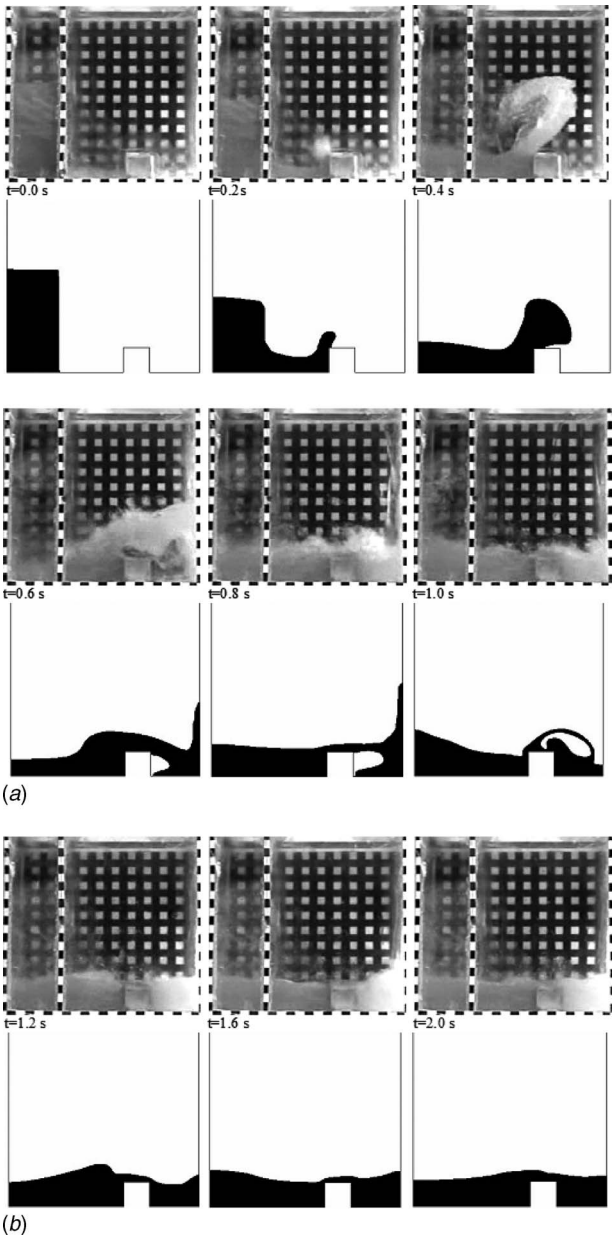



**Fig. 2 Experimental and computed interfaces at different instants for the obstacle with a square cross section. Experimental and computed interfaces at different instants for the obstacle with a square cross section.**

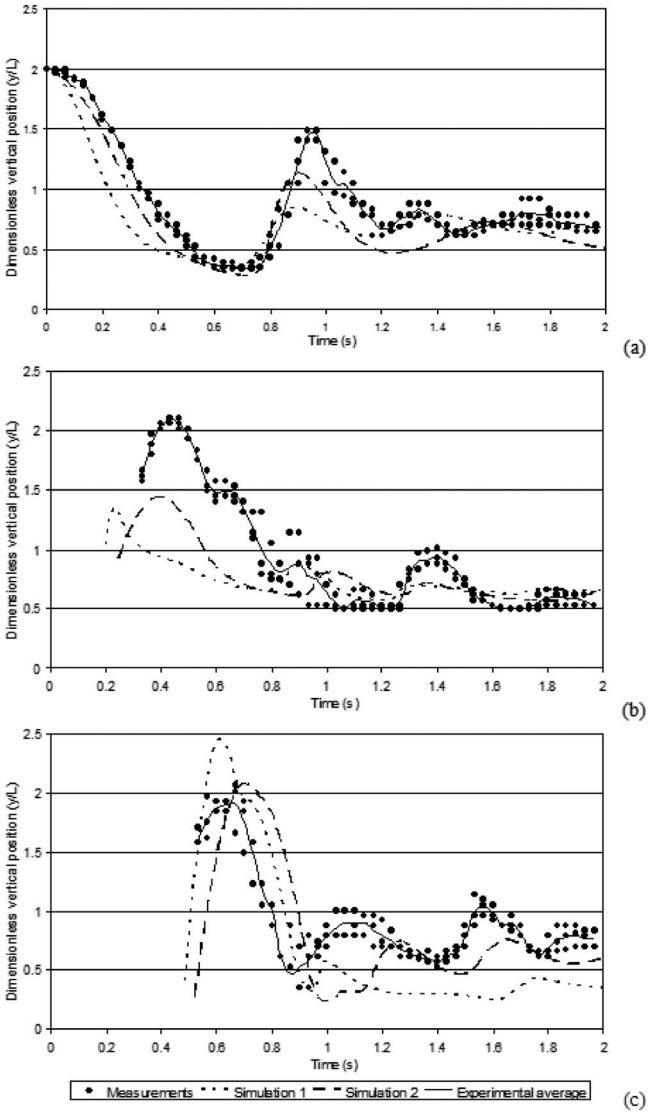



**Fig. 3 Evolution, for the obstacle with a square cross section, of the dimensionless interface vertical position at the left wall (*a*), section at middle of the obstacle (*b*), and the right wall (*c*)**

Two simulations are carried out: Simulation 1 and Simulation 2. In Simulation 1 the gate is assumed to be suddenly removed at time $t=0$ s, while in Simulation 2 the gate is opened not instantaneously but with a finite speed. The opening speed was extracted from the experiments. In Simulation 2, an average opening speed of 0.35 m/s is used. Both simulations include a simple turbulence model defined by Eq. (6) with $C_t=3.57$ and $\mu_{\max}=3.0$ kg/m s$^{-1}$. Moreover, they also both have the DCID defined by Eq. (12) with $C_s=10$.

We note that all the parameters and properties used in these simulations come from the simulations reported in Ref. [15] for cases without obstacles.

**5.1 Obstacle With Square Cross Section.** The interface positions at different instants of Simulation 2 are shown in Fig. 2, together with the corresponding images from the experiment. Since no special model is included to capture the bubbles or drops, the numerical solutions are expected to represent the free surfaces of the fluid bulk in an average sense. As it was reported in Ref. [15], formation of unrealistic bubbles in the fluid bulk is inhibited when the DCID is activated. The results obtained from the present simulations show a satisfactory DCID performance when physical gaps develop in the fluid motion, i.e., those that appear at the back of the obstacles are not removed. The volume of such a gap is not necessarily conserved because the local volume conservation is not activated. Furthermore, we believe that the turbulence model used in the simulations is responsible for the more diffusive behavior exhibited by the numerical predictions in comparison with the experiment. Nevertheless, as it is reported below, there is a reasonably good agreement in representing the evolution of the interface.

Figure 3 shows, for the experiment and Simulations 1 and 2, the evolution of the dimensionless interface vertical position ($y/L$, $y$ being the instantaneous interface vertical position) at the left wall, middle section of the obstacle, and the right wall. The numerical results show trends similar to those obtained from the experiment. Simulation 2, which has a finite gate opening speed, better represents the interface evolution at all the measurement sections. In particular, the train of waves is captured. The computed vertical

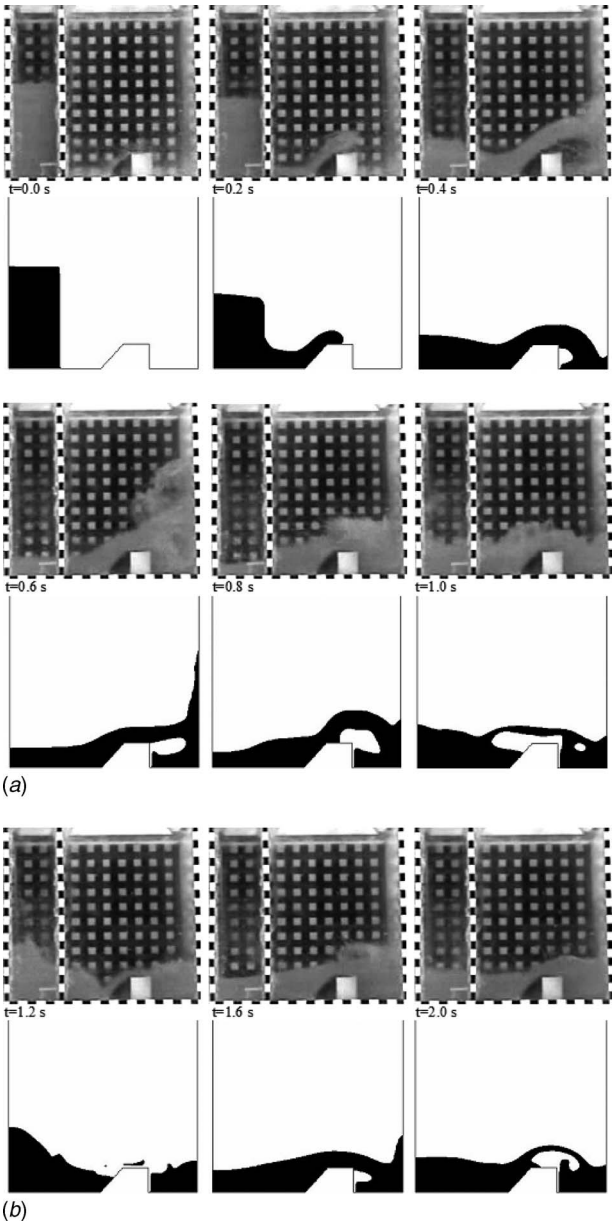



**Fig. 4 Experimental and computed interfaces at different instants for the obstacle with a trapezoidal cross section. Experimental and computed interfaces at different instants for the obstacle with a trapezoidal cross section.**

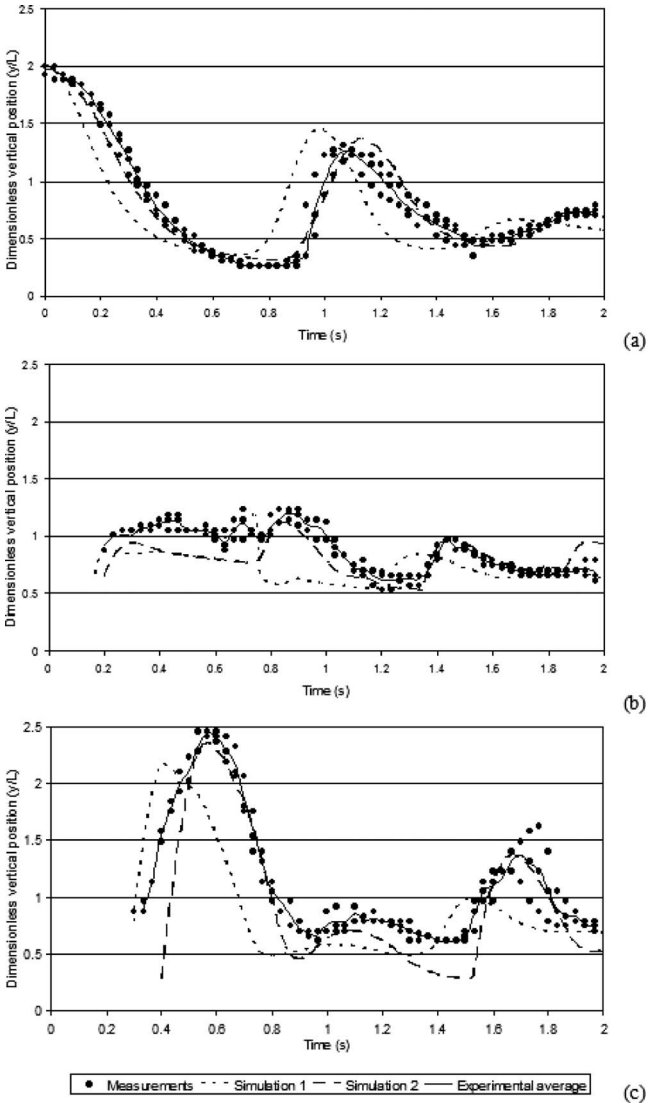



**Fig. 5 Evolution, for the obstacle with a trapezoidal cross section, of the dimensionless interface vertical position at the left wall (*a*), section at middle of the obstacle (*b*), and the right wall (*c*)**

position of the interface is lower than those obtained in the experiment. Compared with the measurements, while the interface evolution is advanced in time at the left wall and the middle section of the obstacle, it is delayed at the right wall. However, the magnitude of first maximum along the right wall is properly represented.

**5.2 Obstacle With Trapezoidal Cross Section.** Figure 4 shows the interfaces at different instants of Simulation 2 and the corresponding images from the experiment. As it was mentioned above, the parameters used in these simulations come from earlier simulations for cases without obstacles. Although at certain instants the free surface exhibits a more diffusive behavior, which can be attributed to those parameters, a reasonably good agreement between the numerical and experimental results is seen particularly for this case. It can be appreciated that the trapezoidal obstacle does not perturb the flow as much as the obstacle with a square cross section.

Figure 5 shows, for the experiment and Simulations 1 and 2, the evolution of the dimensionless interface vertical position at the left wall, middle section of the obstacle, and the right wall. In this case, the results obtained with Simulation 2 are in reasonably good agreement with the experimental data. The computed vertical position of the interface at the left wall practically coincides with the experimental result. The first and second maxima are well represented not only in time but also in magnitude. The numerical behavior of the interface vertical position at the middle of the obstacle also shows good agreement with the experimental results. Although the time at which the interface reaches the right wall is slightly delayed, after that point in time the computed interface vertical position at the wall captures reasonably well the experimental interface evolution.

## 6 Conclusions

The performance of the ETILT was assessed in the simulation of the collapse of a water column over an obstacle with two different geometries. A set of experiments has been carried out to videotape the evolution of the interface.

Simple concepts were included in the model to describe the turbulence effects. The parameters used for characterizing the water behavior were taken from earlier simulations for cases without obstacles. In principle, the values of the properties and parameters could be determined from the present cases of study. Nevertheless, due to the highly complex flow patterns present in the cases with obstacles, the objective function to be minimized could include not only the evolution of the interface position at certain sections of control but also other aspects, e.g., evolution of the air gaps. The characterization of this last aspect is not straightforward. Moreover, the present numerical model is able to capture the formation of the air gaps in an average sense. Hence, we prefer to use the values we set in cases without obstacles. In addition, the results computed with the chosen set of parameters match particularly well the interface evolution obtained for the obstacle of trapezoidal section (which is less dissipative than that with the square obstacle). Therefore we accept that their values can be used to describe the water behavior in the present analyses. In general, maximum water heights and their times of occurrence as well as the time of impingement on the walls and the wave formation are captured reasonably well. The results obtained also demonstrate an acceptable performance from the DCID; e.g., it does not inhibit the gaps that are physical. The effect of the gate opening was found to change the overall behavior of the interface for the range of gate opening speeds considered.

The experimental observations do not exhibit a remarkable 3D behavior. Therefore, a 2D simulation should properly capture the overall flow patterns. Moreover, the experiments exhibit a turbulent response particularly at instants of liquid splashing over walls and obstacles. In the 2D simulations performed in this work, the turbulence model is basically needed to adequately describe the time of liquid impingement over walls and obstacles as well as to reproduce maximum liquid levels and periodicity of waves. The turbulence model adopted depends on the mesh size; hence, its influence decreases when finer meshes are used. However, it helps to include turbulence dissipation effects when we are not carrying out a direct simulation of the turbulent scales.

Overall, the numerical results compare satisfactorily with the data from the experiments.

## Acknowledgment

The authors thank for the support provided by the research projects CONICYT-FONDECYT 1060141 and PCCI 7070012, and by DICYT-USACH.

## References

[1] Tezduyar, T. E., 1991, "Stabilized Finite Element Formulations for Incompressible Flow Computations," Adv. Appl. Mech., **28**, pp. 1–44.

[2] Tezduyar, T., Aliabadi, S., and Behr, M., 1998, "Enhanced-Discretization Interface-Capturing Technique (EDICT) for Computation of Unsteady Flows With Interfaces," Comput. Methods Appl. Mech. Eng., **155**, pp. 235–248.

[3] Koshizuka, S., and Oka, Y., 1996, "Moving-Particle Semi-Implicit Method for Fragmentation of Incompressible Fluid," Nucl. Sci. Eng., **123**, pp. 421–434.

[4] Tezduyar, T. E., 2001, "Finite Element Methods for Flow Problems With Moving Boundaries and Interfaces," Arch. Comput. Methods Eng., **8**, pp. 83–130.

[5] Osher, S., and Fedkiw, P., 2001, "Level Set Methods: An Overview and Some Recent Results," J. Comput. Phys., **169**, pp. 463–502.

[6] Sethian, J. A., 2001, "Evolution, Implementation, and Application of Level Set and Fast Marching Methods for Advancing Fronts," J. Comput. Phys., **169**, pp. 503–555.

[7] Idelsohn, S., Storti, M., and Oñate, E., 2003, "A Lagrangian Meshless Finite Element Method Applied to Fluid-Structure Interaction Problems," Comput. Struct., **81**, pp. 655–671.

[8] Tezduyar, T. E., 2003, "Computation of Moving Boundaries and Interfaces and Stabilization Parameters," Int. J. Numer. Methods Fluids, **43**, pp. 555–575.

[9] Tezduyar, T. E., 2004, "Finite Element Methods for Fluid Dynamics With Moving Boundaries and Interfaces," *Encyclopedia of Computational Mechanics, Volume 3: Fluids*, E. Stein, R. De Borst, and T. J. R. Hughes, eds., Wiley, New York, Chap. 17.

[10] Cueto-Felgueroso, L., Colominas, I., Mosqueiras, G., Navarrina, F., and Casteleiro, M., 2004, "On the Galerkin Formulation of the Smoothed Particle Hydrodynamics Method," Int. J. Numer. Methods Eng., **60**, pp. 1475–1512.

[11] Greaves, D., 2004, "Simulation of Interface and Free Surface Flows in a Viscous Fluid Using Adapting Quadtree Grids," Int. J. Numer. Methods Fluids, **44**, pp. 1093–1117.

[12] Kohno, H., and Tanahashi, T., 2004, "Numerical Analysis of Moving Interfaces Using a Level Set Method Coupled With Adaptive Mesh Refinement," Int. J. Numer. Methods Fluids, **45**, pp. 921–944.

[13] Kulasegaram, S., Bonet, J., Lewis, R. W., and Profit, M., 2004, "A Variational Formulation Based Contact Algorithm for Rigid Boundaries in Two-Dimensional SPH Applications," Comput. Mech., **33**, pp. 316–325.

[14] Cruchaga, M. A., Celentano, D. J., and Tezduyar, T. E., 2005, "Moving-Interface Computations With the Edge-Tracked Interface Locator Technique (ETILT)," Int. J. Numer. Methods Fluids, **47**, pp. 451–469.

[15] Cruchaga, M. A., Celentano, D. J., and Tezduyar, T. E., 2007, "Collapse of a Liquid Column: Numerical Simulation and Experimental Validation," Comput. Mech., **39**, pp. 453–476.

[16] Tezduyar, T. E., 2006, "Interface-Tracking and Interface-Capturing Techniques for Finite Element Computation of Moving Boundaries and Interfaces," Comput. Methods Appl. Mech. Eng., **195**, pp. 2983–3000.

[17] Martin, J., and Moyce, W., 1952, "An Experimental Study of the Collapse of Liquid Columns on a Rigid Horizontal Plane," Philos. Trans. R. Soc. London, Ser. A **244**, pp. 312–324.

# Stabilized Finite Element Methods for the Schrödinger Wave Equation

**Raguraman Kannan**
Graduate Research Assistant
Department of Civil and Materials Engineering,
University of Illinois at Chicago,
Chicago, IL 60607

**Arif Masud**[1]
Associate Professor
Department of Civil and Environmental Engineering,
University of Illinois at Urbana-Champaign,
Urbana, IL 61801
e-mail: amasud@uiuc.edu

*This paper presents two stabilized formulations for the Schrödinger wave equation. First formulation is based on the Galerkin/least-squares (GLS) method, and it sets the stage for exploring variational multiscale ideas for developing the second stabilized formulation. These formulations provide improved accuracy on cruder meshes as compared with the standard Galerkin formulation. Based on the proposed formulations a family of tetrahedral and hexahedral elements is developed. Numerical convergence studies are presented to demonstrate the accuracy and convergence properties of the two methods for a model electronic potential for which analytical results are available.*
[DOI: 10.1115/1.3059564]

*Keywords: Schrödinger wave equation, quantum mechanics, finite elements, stabilized formulations*

## 1 Introduction

The density functional theory (DFT) provides a framework for the calculation of mechanical and electronic properties of materials. Through DFT, the solution of the many-body electronic-structure problem is reduced to a self-consistent solution of the single particle Schrödinger equation. A good overview of the DFT method is presented in Refs. [1,2] and references therein. The time-independent Schrödinger equation, termed as the Schrödinger wave equation (SWE), is a quantum mechanical equation, which is used to determine the electronic structure of periodic solids. SWE has a differential form that involves continuous functions of continuous variables and is therefore suitable for the application of variational methods to the study of electronic properties of periodic materials. The eigensolutions of SWE correspond to different quantum states of the system. Various numerical approaches [2–5] have been adopted for the solution of SWE that include finite element [6–9] and finite difference methods [10,11]. The advantages and utility of finite element method over ab initio methods is discussed in detail in Ref. [7].

In this paper we explore two variational formulations for SWE. Our objective is to study the convergence properties of the finite element methods based on the proposed variational formulations where we have employed lower-order standard Lagrange interpolation functions. We are motivated by the notion of subgrid scale methods [12,13], which in the present context can help in an accurate calculation of higher eigenvalues in the system. Stabilized methods based on variational multiscale ideas, when applied to a number of physical phenomena [14–18] have shown higher accuracy on cruder discretizations as compared with the corresponding standard Galerkin formulations.

An outline of the paper is as follows. Section 2 presents the Schrödinger wave equation and its standard Galerkin form. Section 3 presents the GLS formulation for SWE. Section 4 develops a stabilized formulation that is motivated by the variational multiscale ideas. Section 5 presents results that demonstrate the accuracy and convergence properties of the methods for a model problem (Kronig–Penney problem) for which analytical results are available. Conclusions are drawn in Sec. 6.

## 2 The Schrödinger Wave Equation

Let $\Omega \subset \mathbb{R}^{n_{sd}}$ be an open bounded region with piecewise smooth boundary $\Gamma$. The number of space dimensions $n_{sd}=3$. The Schrödinger wave equation can be written as

$$-\kappa\Delta v(\boldsymbol{r}) - i2\kappa\boldsymbol{k}\cdot\nabla v(\boldsymbol{r}) + \kappa k^2 v(\boldsymbol{r}) + V(\boldsymbol{r})v(\boldsymbol{r}) = \varepsilon(\boldsymbol{k})v(\boldsymbol{r}) \quad \text{in} \quad \Omega \tag{1}$$

The solution of the SWE satisfies Bloch's theorem of periodicity of the wave function. From the periodicity condition, the boundary conditions are taken to be of the form.

$$v(\boldsymbol{r}) = v(\boldsymbol{r}+\boldsymbol{R}) \quad \text{on} \quad \Gamma \tag{2}$$

$$n\cdot\nabla v(\boldsymbol{r}) = n\cdot\nabla v(\boldsymbol{r}+\boldsymbol{R}) \quad \text{on} \quad \Gamma \tag{3}$$

where $v(\boldsymbol{r})$ is the complex valued cell periodic function or the unknown complex scalar field, namely, the wave function (or the eigenfunction) $\boldsymbol{r}$ represents the position vector, $\boldsymbol{n}$ represents the outward unit normal vector to the boundary $\Gamma$ of a unit cell, $V(\boldsymbol{r})$ is the electronic potential or the potential energy of an electron in a charge density $\rho(\boldsymbol{r})$ at the position $\boldsymbol{r}$ and is considered periodic over a unit cell, and $i$ is the imaginary unit. $\varepsilon(\boldsymbol{k})$ is the eigenenergy associated with the particle as a function of wave vector (position vector in reciprocal space) $\boldsymbol{k}$. $\boldsymbol{R}$ refers to the lattice vectors of the unit cell, and $\kappa=\hbar^2/2m$ and $\hbar=h/2\pi$ are constants, where $\hbar$ is Planck's constant and $m$ is the effective mass of electron.

*Remark 1*. The values of $V(\boldsymbol{r})$ and $v(\boldsymbol{r})$ in a periodic solid are completely determined by their values in a single unit cell. Therefore solutions of the Schrödinger equation in a periodic solid can be reduced to their solutions in a single unit cell, subject to periodic boundary conditions consistent with Eqs. (2) and (3), respectively.

**2.1 The Standard Weak Form.** Let $\mathcal{V}\subset \mathrm{H}^1(\Omega^{n_{sd}}) \cap \mathrm{C}^0(\Omega^{n_{sd}})$ denote the space of trial solutions and weighting functions for the unknown scalar field where periodicity of the boundary condition is embedded in the admissible space.

[1]Corresponding author.

Contributed by the Applied Mechanics Division of ASME for publication in the JOURNAL OF APPLIED MECHANICS. Manuscript received November 27, 2007; final manuscript received July 10, 2008; published online January 14, 2009. Review conducted by Tayfun E. Tezduyar.

$$\mathcal{V} = \{v | v \in \mathrm{H}^1(\Omega^{n_{\mathrm{sd}}}),\ v(\boldsymbol{r}) = v(\boldsymbol{r}+\boldsymbol{R})\}, \quad \forall\, \boldsymbol{r} \in \Gamma \tag{4}$$

The standard weak form for the complex valued problem is

$$-(w, i2\kappa \boldsymbol{k}\cdot\nabla v) + (\nabla w, \kappa \nabla v) + (w, (\kappa k^2 + V)v) = (w, \varepsilon v) \tag{5}$$

where $w$ is the weighting function for $v$, and $(\cdot,\cdot)=\int_\Omega(\cdot)d\Omega$, i.e., $L_2$ product of the indicated arguments over domain $\Omega$. Discretization of the standard weak form gives rise to a generalized eigenvalue problem for the complex valued cell periodic function or the eigenfunction $\nu(\boldsymbol{r})$ and the associated eigenenergy $\varepsilon(\boldsymbol{k})$.

*Remark 2*. Galerkin method seems to work for the present problem, however typical applications in the literature have been presented in the context of hermite cubic functions [6,7]. Employing lower-order Lagrange shape functions in the standard Galerkin formulation results in reduced accuracy in the evaluation of higher eigenvalues in the system.

*Remark 3*. Our objective in this work is to explore numerical methods that can provide higher accuracy in the estimation of higher eigenvalues, while using lower-order Lagrange shape functions on computational domains that are less dense than the grids employed for the corresponding Galerkin method.

## 3 The Galerkin/Least-Square Stabilized Form

This section presents the GLS form for the Schrödinger wave equation. GLS stabilization is a standard technique employed in computational fluid dynamics to enhance the stability of the underlying Galerkin variational formulations, which also manifests itself in terms of improved accuracy on relatively cruder discretizations. The basic idea of stabilized methods is to add a least-squares form of the Euler–Lagrange equations to the standard Galerkin form presented in Eq. (5), thus strengthening the variational structure of the problem.

$$\begin{aligned}(\nabla w, \kappa\nabla v) - (w, i2\kappa\boldsymbol{k}\cdot\nabla v) + (w, (\kappa k^2 + V - \varepsilon)v) + ((-\kappa\Delta \\ - i2\kappa\boldsymbol{k}\cdot\nabla + \kappa k^2 + V)w, \tau^{\mathrm{GLS}}[(-\kappa\Delta - i2\kappa\boldsymbol{k}\cdot\nabla + \kappa k^2 + V \\ - \varepsilon)v]) = 0\end{aligned} \tag{6}$$

In Eq. (6) we have used the idea of Petrov–Galerkin methods and have dropped the $\varepsilon$ term in the weighting function slot of the additional stabilization term. This helps in reducing the order of the resulting eigenvalue problem from quadratic to linear. In Eq. (6) $\tau^{\mathrm{GLS}}$ is the stabilization parameter that will be defined later.

*Remark 4*. The GLS method is shown to yield higher accuracy for many physical problems [12,19] and in the present case it sets the stage for exploring the variational multiscale ideas for application to SWE.

## 4 The Variational Multiscale Method

This section develops and explores the properties of another stabilized method that finds its roots in the variational multiscale method proposed by Hughes [12] and we term it as the Hughes Variational Multiscale (HVM) form. The basic premise of multiscale approach is to acknowledge the presence of the fine-scales that may not be resolved by a given spatial discretization. We consider the bounded domain $\Omega$ to be discretized into nonoverlapping regions $\Omega^e$ (element domains) with boundaries $\Gamma^e$, $e=1$, and $2\cdots n_{\mathrm{umel}}$ such that $\Omega=\cup_{e=1}^{n_{\mathrm{umel}}}\bar{\Omega}^e$. We denote the union of element interiors and element boundaries by $\Omega'$ and $\Gamma'$. respectively, i.e., $\Omega'=\cup_{e=1}^{n_{\mathrm{umel}}}(\mathrm{int})\Omega^e$ (element interiors) and $\Gamma'=\cup_{e=1}^{n_{\mathrm{umel}}}\Gamma^e$ (element boundaries) We assume an overlapping sum decomposition of the scalar field $v(\boldsymbol{r})$ into coarse- or resolvable-scales and fine- or the subgrid-scales.

$$v(\boldsymbol{x}) = \bar{v}(\boldsymbol{x}) + v'(\boldsymbol{x}) \tag{7}$$

Likewise, we assume an overlapping sum decomposition of the weighting function into the coarse- and the fine-scale components, respectively.

$$w(\boldsymbol{x}) = \bar{w}(\boldsymbol{x}) + w'(\boldsymbol{x}) \tag{8}$$

We further make an assumption that the subgrid scales, although nonzero within the elements, vanish identically over the element boundaries, i.e., $v'=w'=0$ on $\Gamma'$.

We now introduce the appropriate spaces of functions for the coarse- and fine-scale fields and specify direct sum decomposition on these spaces, i.e., $\mathcal{V}=\bar{\mathcal{V}}\oplus\mathcal{V}'$ where $\bar{\mathcal{V}}$ is the space of trial solutions and weighting functions for the coarse-scale field and is identified with the standard finite element space, while $\mathcal{V}'$ is the space of fine-scale functions. These spaces are subject to the restriction imposed by the stability of the formulation that requires $\bar{\mathcal{V}}$ and $\mathcal{V}'$ to be linearly independent.

**4.1 The Multiscale Variational Problem.** We now substitute the trial solutions (7) and the weighting functions (8) in the standard variational form (5), which yields

$$\begin{aligned}-(\bar{w}+w', i2\kappa\boldsymbol{k}\cdot\nabla(\bar{v}+v')) + (\nabla(\bar{w}+w'), \kappa\nabla(\bar{v}+v')) + (\bar{w} \\ + w', (\kappa k^2 + V)(\bar{v}+v')) = (\bar{w}+w', \varepsilon(\bar{v}+v'))\end{aligned} \tag{9}$$

With suitable assumptions on the fine-scale field (i.e., fine-scales vanish at the interelement boundaries) and employing the linearity of the weighting function slot, we can split the problem into coarse- and fine-scale parts, indicated as $\bar{\mathcal{W}}$ and $\mathcal{W}'$, respectively.

The coarse-scale problem $\bar{\mathcal{W}}$,

$$\begin{aligned}-(\bar{w}, i2\kappa\boldsymbol{k}\cdot\nabla(\bar{v}+v')) + (\nabla\bar{w}, \kappa\nabla(\bar{v}+v')) + (\bar{w}, (\kappa k^2 + V)(\bar{v} \\ + v')) = (\bar{w}, \varepsilon(\bar{v}+v'))\end{aligned} \tag{10}$$

The fine-scale problem $\mathcal{W}'$,

$$\begin{aligned}-(w', i2\kappa\boldsymbol{k}\cdot\nabla(\bar{v}+v')) + (\nabla w', \kappa\nabla(\bar{v}+v')) + (w', (\kappa k^2 + V)(\bar{v} \\ + v')) = (w', \varepsilon(\bar{v}+v'))\end{aligned} \tag{11}$$

The underlying idea at this point is to solve the fine-scale problem (11), which is defined over the sum of element interiors, to obtain the fine-scale solution $v'$. This solution is then substituted in the coarse-scale problem given by Eq. (10), thereby eliminating the fine-scales, yet retaining their effect.

**4.2 Solution of the Fine-Scale Problem ($\mathcal{W}'$).** Employing linearity of the solution slot in Eq. (11), applying integration by parts, and rearranging terms, the fine-scale problem reduces to

$$\begin{aligned}-(w', i2\kappa\boldsymbol{k}\cdot\nabla v')_{\Omega'} + (\nabla w', \kappa\nabla v')_{\Omega'} + (w', (\kappa k^2 + V)v')_{\Omega'} \\ - (w', \varepsilon v')_{\Omega'} = (w', i2\kappa\boldsymbol{k}\cdot\nabla\bar{v} + \kappa\Delta\bar{v} - (\kappa k^2 + V)\bar{v} + \varepsilon\bar{v})_{\Omega'}\end{aligned} \tag{12}$$

From Eq. (12) one can see that the fine-scale problem is driven by the residual of Euler–Lagrange equations of the coarse scales defined over the sum of element interiors. Without loss of generality, we assume that the fine-scales $v'$ and $w'$ are represented via bubbles over element domains, that is,

$$v'|_{\Omega^e} = b_1^e v'_e \quad \text{on} \quad \Omega^e \tag{13}$$

$$w'|_{\Omega^e} = b_2^e w'_e \quad \text{on} \quad \Omega^e \tag{14}$$

where $b_1^e$ and $b_2^e$ represent the bubble shape functions, and $v'_e$ and $w'_e$ represent the coefficients for the fine-scale trial solutions and weighting functions, respectively. Substituting Eqs. (13) and (14) in the fine-scale problem (12) we get

$$\begin{aligned}-(b_2^e w'_e, i2\kappa\boldsymbol{k}\cdot\nabla b_1^e v'_e)_{\Omega'} + (\nabla b_2^e w'_e, \kappa\nabla b_1^e v'_e)_{\Omega'} + (b_2^e w'_e, (\kappa k^2 \\ + V)b_1^e v'_e)_{\Omega'} - (b_2^e w'_e, \varepsilon b_1^e v'_e)_{\Omega'} = (b_2^e w'_e, i2\kappa\boldsymbol{k}\cdot\nabla\bar{v} + \kappa\Delta\bar{v} \\ - (\kappa k^2 + V)\bar{v} + \varepsilon\bar{v})_{\Omega'}\end{aligned} \tag{15}$$

Taking the constant coefficients $w'_e$ and $v'_e$ out of the integral expressions and employing arbitrariness of $w'_e$, we can solve for the fine-scale coefficients $v'_e$.

$$v'_e = \frac{-1(b_2^e,(-\kappa\Delta - i2\kappa\boldsymbol{k}\cdot\nabla + \kappa k^2 + V - \varepsilon)\bar{v})_{\Omega'}}{[(\nabla b_2^e, \kappa\nabla b_1^e)_{\Omega'} + (b_2^e,(-i2\kappa\boldsymbol{k}\cdot\nabla + \kappa k^2 + V - \varepsilon)b_1^e)_{\Omega'}]} \tag{16}$$

We can now reconstruct the fine-scale field via recourse to Eq. (13). In order to keep the presentation simple, and for the case where the residual of the coarse scales over element interiors can be considered constant, we can simplify fine-scales $v'(\boldsymbol{x})$ as follows:

$$v'(\boldsymbol{x}) = -\tau[(-\kappa\Delta - i2\kappa\boldsymbol{k}\cdot\nabla + \kappa k^2 + V - \varepsilon)\bar{v}] \tag{17}$$

Within the context of stabilized methods, $\tau$ is defined as the stability parameter. In the derivation presented above $\tau$ has an explicit form

$$\tau = b_1^e \int_{\Omega^e} b_2^e d\Omega[(\nabla b_2^e, \kappa\nabla b_1^e)_{\Omega'} + (b_2^e,(-i2\kappa\boldsymbol{k}\cdot\nabla + \kappa k^2 + V - \varepsilon)b_1^e)_{\Omega'}]^{-1} \tag{18}$$

*Remark 5.* In our numerical calculations we have simplified the definition of $\tau$ by setting $\varepsilon=0$ in Eq. (18).

*Remark 6.* The definition of the bubble functions completely resides in the definition of the stability parameter $\tau^{\mathrm{HVM}}$. Consequently, a choice of specific bubbles only affects the value of $\tau^{\mathrm{HVM}}$. Stabilization parameters that are based on element-level matrices and element-level vectors have also been used in the Streamline Upwind Petrov–Galerkin (SUPG) and GLS methods [19].

**4.3 The Coarse Scale Problem ($\overline{\mathcal{VW}}$).** Employing linearity of the solution slot in the coarse-scale subproblem (10) and applying integration by parts, one can combine $v'$ terms as

$$-(\bar{w},i2\kappa\boldsymbol{k}\cdot\nabla\bar{v}) + (\nabla\bar{w},\kappa\nabla\bar{v}) + (\bar{w},(\kappa k^2 + V - \varepsilon)\bar{v}) + ((i2\kappa\boldsymbol{k}\cdot\nabla - \kappa\Delta + \kappa k^2 + V - \varepsilon)\bar{w},v') = 0 \tag{19}$$

Substituting $v'$ from Eq. (17) in Eq. (19) yields the resulting stabilized formulation.

$$(\nabla\bar{w},\kappa\nabla\bar{v}) - (\bar{w},i2\kappa\boldsymbol{k}\cdot\nabla\bar{v}) + (\bar{w},(\kappa k^2 + V - \varepsilon)\bar{v}) - ((-\kappa\Delta + i2\kappa\boldsymbol{k}\cdot\nabla + \kappa k^2 + V - \varepsilon)\bar{w},\tau[(-\kappa\Delta - i2\kappa\boldsymbol{k}\cdot\nabla + \kappa k^2 + V - \varepsilon)\bar{v}]) = 0 \tag{20}$$

**4.4 The HVM Stabilized Form.** The HVM stabilized form (20) is completely expressed in terms of the coarse or resolvable-scales. Therefore, in order to keep the notation simple we drop the superposed bars and we write the resulting form as

$$(\nabla w,\kappa\nabla v) - (w,i2\kappa\boldsymbol{k}\cdot\nabla v) + (w,(\kappa k^2 + V - \varepsilon)v) - ((-\kappa\Delta + i2\kappa\boldsymbol{k}\cdot\nabla + \kappa k^2 + V - \varepsilon)w,\tau[(-\kappa\Delta - i2\kappa\boldsymbol{k}\cdot\nabla + \kappa k^2 + V - \varepsilon)v]) = 0 \tag{21}$$

*Remark 7.* The first three terms in Eq. (21) are the standard Galerkin terms. The fourth term has appeared due to the assumption of the existence of fine-scales. This term is not present in the standard Galerkin formulation.

*Remark 8.* The subgrid scales are proportional to the residual of the coarse scales as shown in Eqs. (12) and (17), i.e., it is a residual based method and therefore satisfies consistency ab initio.

*Remark 9.* When compared with the standard Galerkin method, the multiscale approach involves additional integrals that are evaluated elementwise and represent the effects of the subgrid scales that are modeled in terms of the residuals of the coarse scales of the problem.

*Remark 10.* For the numerical solution of the variational problem where the periodic Dirichlet and Neumann boundary conditions presented in Eqs. (1) and (2) are already embedded in Eq. (21), we employ the procedure outlined in Refs. [6–8] and modify element connectivity to produce value-periodic basis functions.

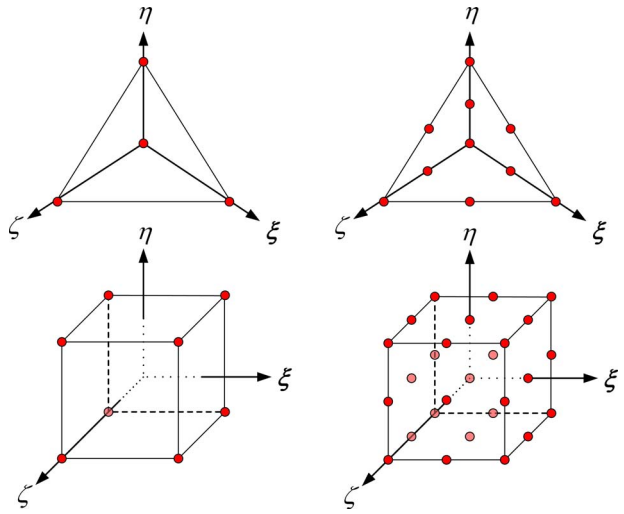



**Fig. 1 A family of 3D linear and quadratic elements**

**4.5 Quadratic Eigenvalue Problem for the HVM Form.** The solution procedure for the HVM form (21) involves a quadratic eigenvalue problem described as follows:

$$(\varepsilon^2\boldsymbol{M} + \varepsilon\boldsymbol{C} + \boldsymbol{K})\boldsymbol{x} = 0 \tag{22}$$

where $\boldsymbol{M}$, $\boldsymbol{C}$, and $\boldsymbol{K}$ are $n\times n$ matrices, $\varepsilon$ is the scalar eigenvalue, and $\boldsymbol{x}$ is the eigenvector. In order to solve this problem one has to linearize it as follows:

$$\boldsymbol{A}\boldsymbol{z} = \varepsilon\boldsymbol{B}\boldsymbol{z} \tag{23}$$

where

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{K} & -\boldsymbol{C} \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M} \end{bmatrix}, \quad \text{and} \quad \boldsymbol{z} = \begin{bmatrix} \boldsymbol{x} \\ \varepsilon\boldsymbol{x} \end{bmatrix} \tag{24}$$

*Remark 11.* The HVM eigenvalue problem increases the size of the matrices from $n\times n$ to $2n\times 2n$, which also increases the cost of computation.

## 5 Numerical Examples

Figure 1 shows a family of 3D elements that consist of 4- and 10-node tetrahedra and 8- and 27-node brick elements for the numerical solution of the problem. In the numerical tests presented in this section, the functional form of $\tau^{\mathrm{GLS}}$ is taken to be the same as that of $\tau^{\mathrm{GLS}}$, which is defined in Eq. (18). The bubble functions employed for the evaluation of $\tau$ are at least one order higher than the functions employed for the complex valued wave function. Accordingly, quadratic and cubic bubble functions were used for the 8-node and 27-node brick elements, respectively. In the case of both linear and quadratic tetrahedral elements, quadratic bubbles were used as this bubble function enriches the space of functions in both the cases.

We present the convergence study for the 3D generalized Kronig–Penney problem. The domain under consideration is a cube with electronic potential $V(\boldsymbol{r})$ given by

$$V(\boldsymbol{r}) = V_{1D}(x) + V_{1D}(y) + V_{1D}(z) \quad \text{in} \quad \Omega \tag{25}$$

where

$$V_{1D}(s) = \begin{cases} 0 & 0 \le s < 2 \ \text{a.u.} \\ 6.5\ \text{Ry} & 2 \le s < 3 \ \text{a.u.} \end{cases}$$

Figures 2–9 present convergence rates for the fractional error in the first, fifth, and seventh eigenvalues for the Galerkin, GLS, and HVM methods with linear and quadratic shape functions at a selected but otherwise arbitrary $\boldsymbol{k}$ point. The theoretical convergence

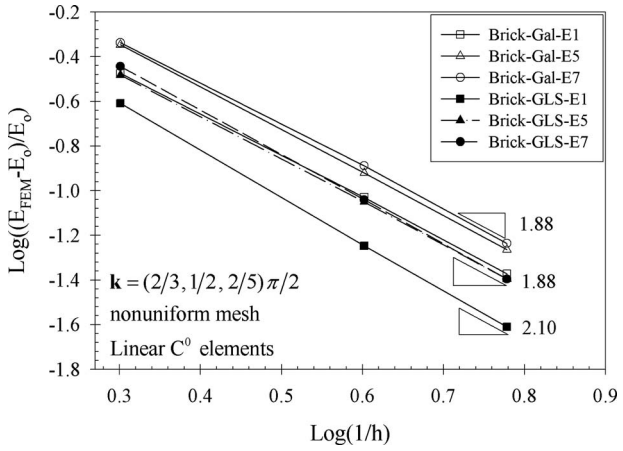



**Fig. 2 Convergence rates for eigenvalues using linear brick elements (GLS)**

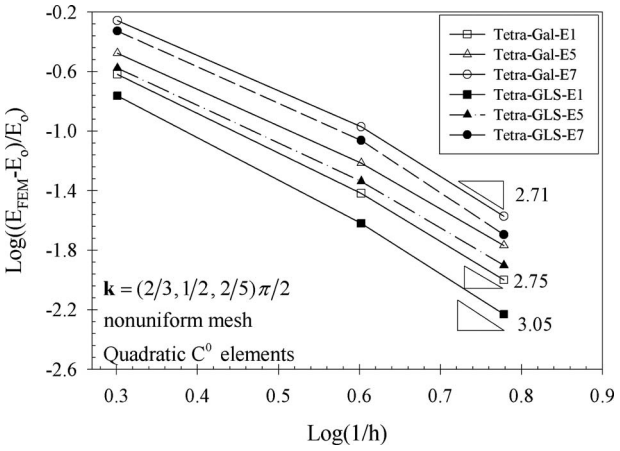



**Fig. 5 Convergence rates for eigenvalues using quadratic tetrahedral elements (GLS)**

rate for the eigenvalues for linear and quadratic elements is $k+1$, where $k$ is the order for the interpolation of the complex valued wave function $v$. Computed rates corroborate the theoretical predictions [20]. In each of the test cases the $L_2$ error in the computed eigenvalues is smallest for the first eigenvalue and it successively increases for the higher eigenvalues. In these test cases Galerkin solution is the least accurate for any given mesh.

**5.1 Convergence Rate Results for the GLS Stabilized Formulation.** Figures 2–5 show convergence properties for the GLS method. Meshes employed for the linear elements are composed of $4\times4\times4$, $8\times8\times8$, and $12\times12\times12$ elements, while meshes employed for the quadratic elements are composed of 2

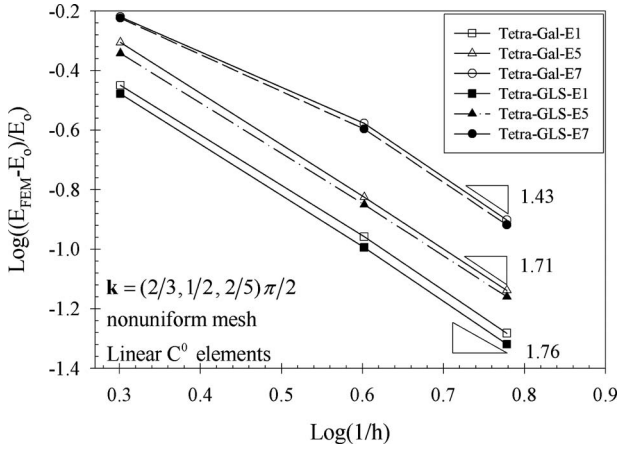



**Fig. 3 Convergence rates for eigenvalues using linear tetrahedral elements (GLS)**

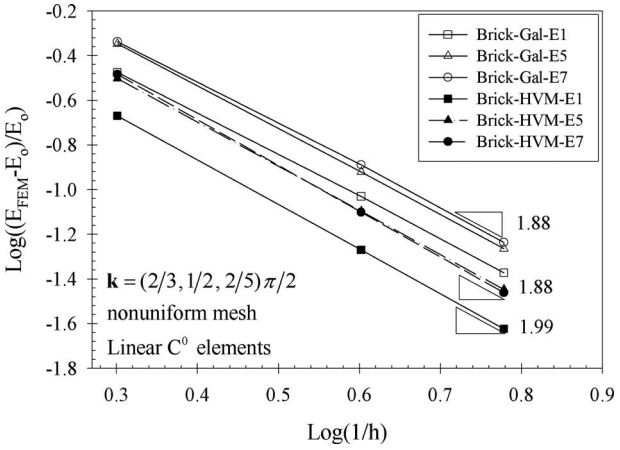



**Fig. 6 Convergence rates for eigenvalues using linear brick elements (HVM)**

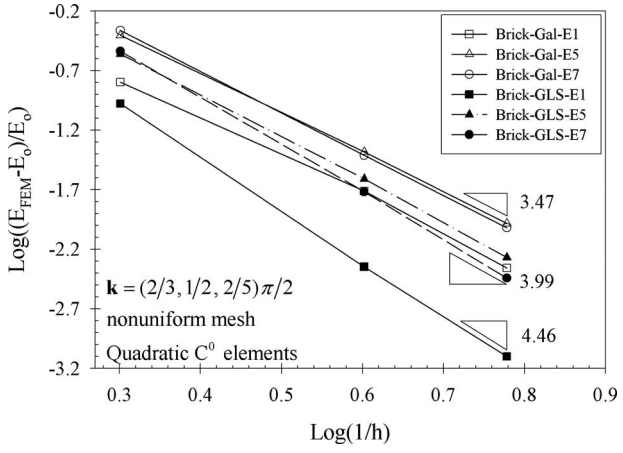



**Fig. 4 Convergence rates for eigenvalues using quadratic brick elements (GLS)**

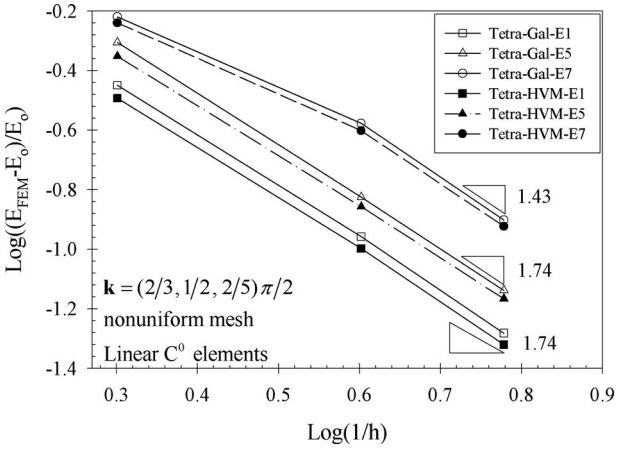



**Fig. 7 Convergence rates for eigenvalues using linear tetrahedral elements (HVM)**

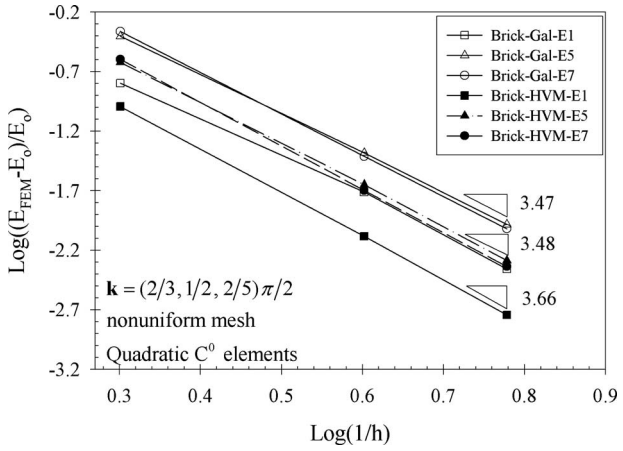



**Fig. 8 Convergence rates for eigenvalues using quadratic brick elements (HVM)**

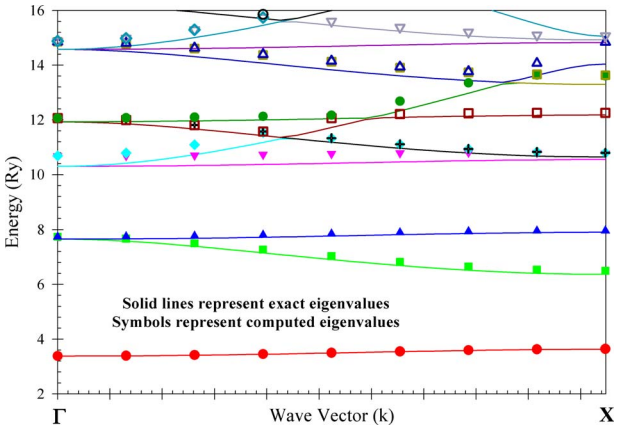



**Fig. 10 Energy band diagram for the GLS formulation**

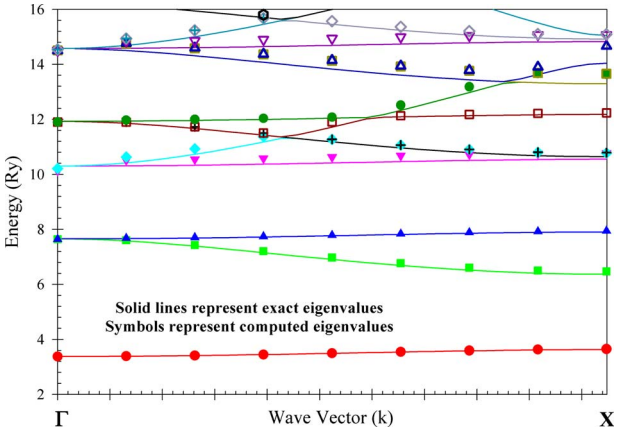



**Fig. 11 Energy band diagram for the HVM formulation**

$\times 2 \times 2$, $4 \times 4 \times 4$, and $6 \times 6 \times 6$ elements. Figures 2 and 3 show a quadratic convergence rate for the computed eigenvalues for linear elements, while cubic convergence rate is attained for the quadratic elements, as shown in Figs. 4 and 5. In all the cases although there is no increase in convergence rates for the GLS stabilized method as compared with the standard Galerkin method, the results clearly show that the GLS eigenvalues are more accurate than those obtained via the standard Galerkin method.

**5.2 Convergence Rate Results for the HVM Formulation.** Figures 6–9 show convergence rates for the HVM method. Meshes employed for the linear elements are composed of $6 \times 6 \times 6$, $9 \times 9 \times 9$, and $12 \times 12 \times 12$ elements, while meshes employed for the quadratic elements are composed of $2 \times 2 \times 2$, $4 \times 4 \times 4$, and $6 \times 6 \times 6$ elements. Once again optimal convergence rates are attained in all the test cases.

**5.3 Energy Band Diagram.** Figures 10 and 11 show the eigenvalues computed via the GLS and the HVM formulations for the $4 \times 4 \times 4$ quadratic brick mesh. Solid lines show the analytical solution and the circles correspond to the computed values. Any interested reader is referred to Chapters 2 and 3 of Ref. [21] for a description of the band diagram and the Brillouin zone. In case of Kronig–Penney problem, the first Brillouin zone is a cube of length $2\pi/3$. In Fig. 10 $\Gamma$ represents the center of the first Brillouin zone and $\mathbf{X}$ represents the center of the face of the first Brillouin zone with unit normal vector $\langle 1,0,0 \rangle$.

**5.4 Convergence Rate for a High Value of the Electronic Potential.** The range of values for the pseudopotential typically lies between $-60$ Ry and $-10$ Ry units. Therefore tests were carried out to see the effects of higher values of the potentials. Figures 12–19 show convergence of the fractional error in the eigenvalues for $V = 60.5$ Ry. Meshes employed for the present

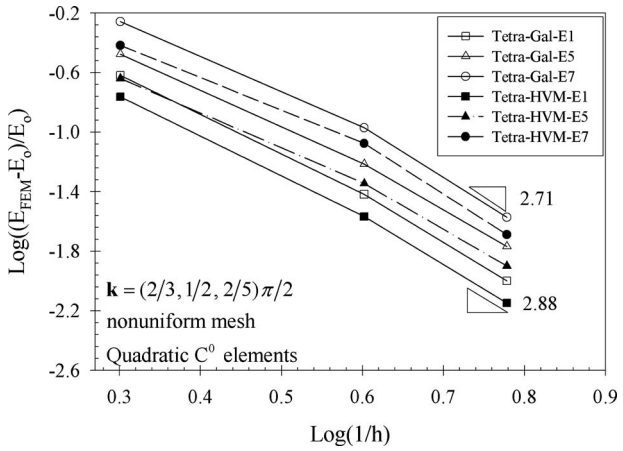



**Fig. 9 Convergence rate for eigenvalues using quadratic tetrahedral elements (HVM)**

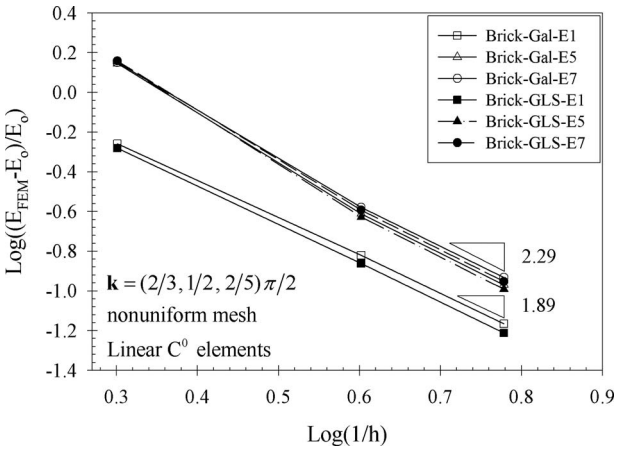



**Fig. 12 Convergence rates for eigenvalues using linear brick elements (GLS)**

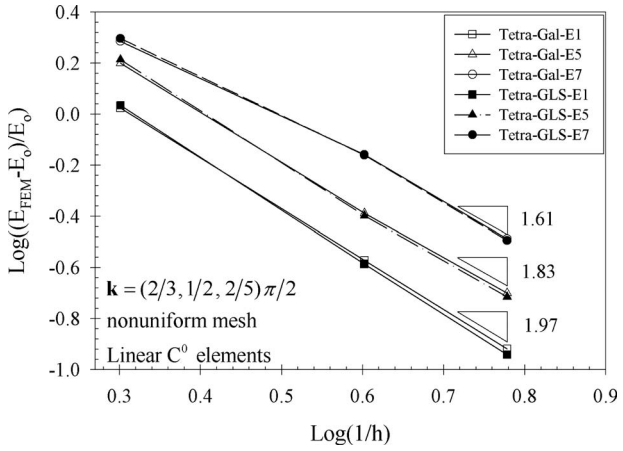



**Fig. 13 Convergence rates for eigenvalues using linear tetrahedral elements (GLS)**

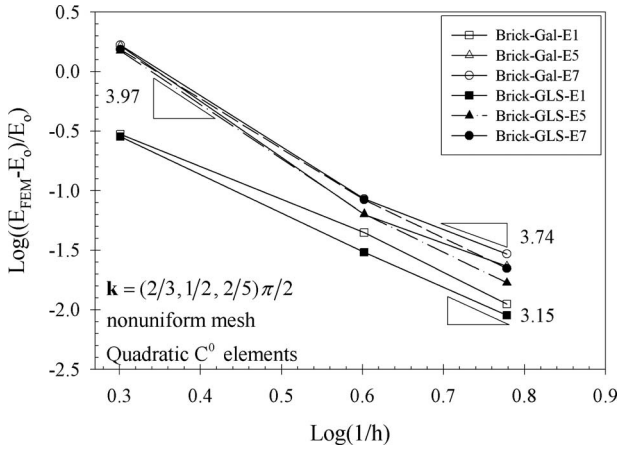



**Fig. 14 Convergence rates for eigenvalues using quadratic brick elements (GLS)**

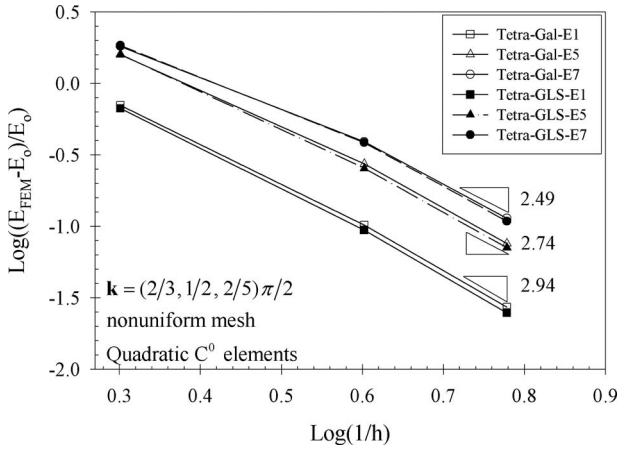



**Fig. 15 Convergence rates for eigenvalues using quadratic tetrahedral elements (GLS)**

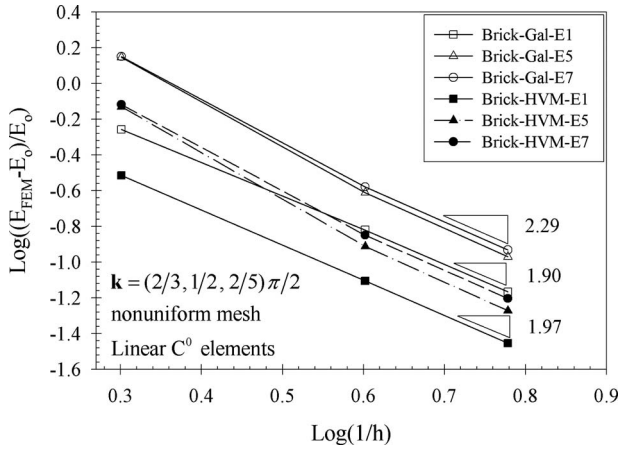



**Fig. 16 Convergence rates for eigenvalues using linear brick elements (HVM)**

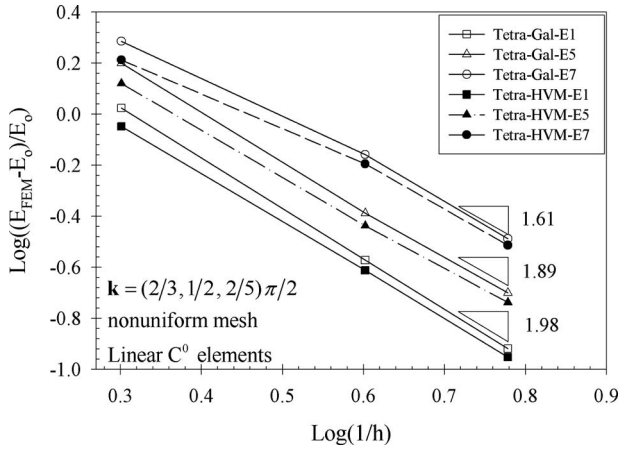



**Fig. 17 Convergence rates for eigenvalues using linear tetrahedral elements (HVM)**

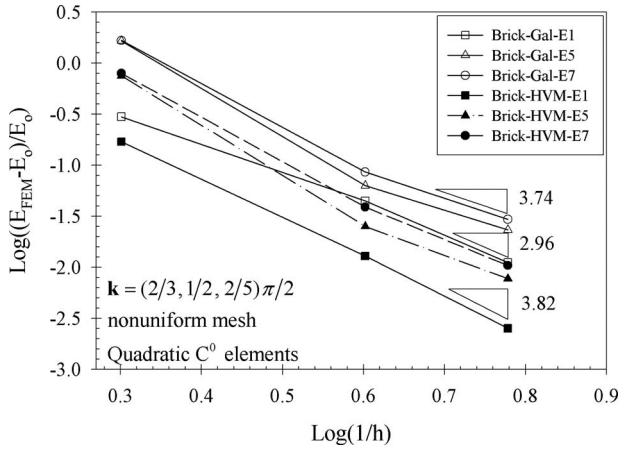



**Fig. 18 Convergence rates for eigenvalues using quadratic brick elements (HVM)**

study are the same as the ones used in Secs. 5.1 and 5.2. Once again optimal rates in the norms considered are attained for the various test cases. The normalized error for Galerkin method is higher even for the first few eigenvalues, as compared with the GLS and the HVM methods.

## 6 Concluding Remarks

We have presented two finite element formulations for the solution of the Schrödinger wave equation: (a) the GLS formulation and (b) the HVM formulation. The GLS formulation when reduced to the standard eigenvalue problem, yields solution at a computational cost that is comparable to that of the Galerkin method, however with higher accuracy in the evaluation of the

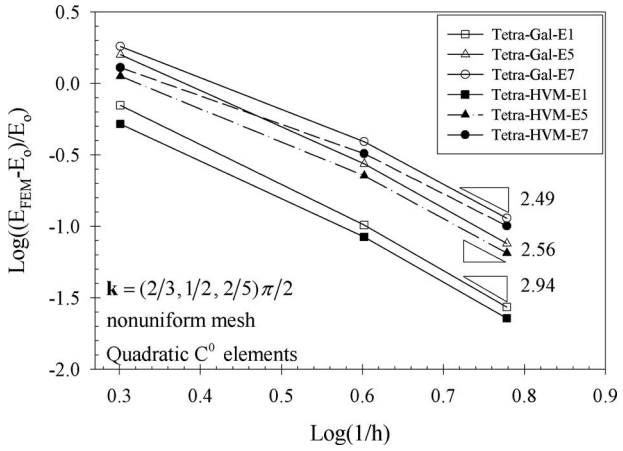



**Fig. 19 Convergence rates for eigenvalues using quadratic tetrahedral elements (HVM)**

higher eigenvalues as compared with the Galerkin method. The HVM formulation also yields optimal convergence rates; however, it leads to a quadratic eigenvalue problem that adds to the cost of computation. The numerical convergence rates of the methods are investigated via the Kronig–Penney problem that serves as a benchmark test case for investigating the mathematical properties of the methods. The quadratic elements show a substantial gain in accuracy as compared with the 3D linear elements. Among the quadratic elements, quadratic bricks show better accuracy as compared with the quadratic tetrahedral element.

## Acknowledgment

This work was supported by the Office of Naval Research (Grant No. N000014-00-1-0687) and the National Academies grant (Grant No. NAS 7251-05-005). This support is gratefully acknowledged.

## References

[1] Chermette, H., 1998, "Density Functional Theory: A Powerful Tool for Theoretical Studies in Coordination Chemistry," Coord. Chem. Rev., **178–180**, pp. 699–721.
[2] Martin, R. M., 2004, *Electronic Structure: Basic Theory and Practical Methods*, Cambridge University Press, Cambridge.
[3] Liu, B., Jiang, H., Johnson, H. T., and Huang, Y., 2004, "The Influence of Mechanical Deformation on the Electrical Properties of Single Wall Carbon Nanotubes," J. Mech. Phys. Solids, **52**(1), pp. 1–26.
[4] Qian, D., Wagner, G. J., Liu, W. K., Yu, M., and Ruoff, R. S., 2002, "Mechanics of Carbon Nanotubes," Appl. Mech. Rev., **55**, pp. 495–533.
[5] Liu, W. K., Karpov, E. G., Zhang, S., and Park, H. S., 2004, "An Introduction to Computational Nanomechanics and Materials," Comput. Methods Appl. Mech. Eng., **193**, pp. 1529–1578.
[6] Pask, J. E., Klein, B. M., Sterne, P. A., and Fong, C. Y., 2001, "Finite-Element Methods in Electronic-Structure Theory," Comput. Phys. Commun., **135**, pp. 1–34.
[7] Pask, J. E., 1999, "A Finite-Element Method for Large-Scale Ab Initio Electronic-Structure Calculations," Ph.D. thesis, University of California, Davis.
[8] Pask, J. E., Klein, B. M., Fong, C. Y., and Sterne, P. A., 1999, "Real-Space Local Polynomial Basis for Solid-State Electronic-Structure Calculations: A Finite-Element Approach," Phys. Rev. B, **59**, pp. 12352–12358.
[9] Jun, S., and Liu, W. K., 2007, "Moving Least Square Basis for Band-Structure Calculations of Natural and Artificial Crystals," *Material Substructure in Complex Bodies: From Atomic Level to Continuum*, 1st ed., G. Capriz and P. M. Mariano, eds., Elsevier, Amsterdam, pp. 163–205.
[10] Chelikowsky, J. R., Troullier, N., and Saad, Y., 1994, "Finite-Difference-Pseudopotential Method: Electronic Structure Calculations Without a Basis," Phys. Rev. Lett., **72**, pp. 1240–1243.
[11] Chelikowsky, J. R., Troullier, N., Wu, K., and Saad, Y., 1994, "Higher-Order Finite-Difference Pseudopotential Method: An Application to Diatomic Molecules," Phys. Rev. B, **50**, pp. 11355–11364.
[12] Hughes, T. J. R., 1995, "Multiscale Phenomena: Green's Functions, The Dirichlet-to-Neumann Formulation, Subgrid Scale Models, Bubbles and the Origins of Stabilized Methods," Comput. Methods Appl. Mech. Eng., **127**, pp. 387–401.
[13] Masud, A., and Franca, L. P., 2008, "A Hierarchical Multiscale Framework for Problems With Multiscale Source Terms," Comput. Methods Appl. Mech. Eng., **197**(33–40), pp. 2692–2700.
[14] Masud, A., and Hughes, T. J. R., 2002, "A Stabilized Mixed Finite Element Method for Darcy Flow," Comput. Methods Appl. Mech. Eng., **191**, pp. 4341–4370.
[15] Masud, A., and Khurram, R., 2004, "A Stabilized/Multiscale Method for the Advection-Diffusion Equation," Comput. Methods Appl. Mech. Eng., **192**(13–14), pp. 1–24.
[16] Masud, A., and Bergman, L. A., 2005, "Application of Multiscale Finite Element Methods to the Solution of the Fokker–Planck Equation," Comput. Methods Appl. Mech. Eng., **194**, pp. 1513–1526.
[17] Masud, A., and Khurram, R., 2005, "A Multiscale Finite Element Method for the Incompressible Navier–Stokes Equations," Comput. Methods Appl. Mech. Eng., **194**(35), pp. 16–42.
[18] Masud, A., and Xia, K., 2006, "A Variational Multiscale Method for Computational Inelasticity: Application to Superelasticity in Shape Memory Alloys," Comput. Methods Appl. Mech. Eng., **195**, pp. 4512–4531.
[19] Tezduyar, T. E., and Osawa, Y., 2000, "Finite Element Stabilization Parameters Computed from Element Matrices and Vectors," Comput. Methods Appl. Mech. Eng., **190**, pp. 411–430.
[20] Strang, G., and Fix, G. J., 1973, *An Analysis of the Finite Element Method*, Prentice-Hall, Englewood Cliffs, NJ.
[21] Pierret, R. F., 2003, *Advanced Semiconductor Fundamentals*, Vol. 6, Prentice-Hall, Englewood Cliffs, NJ.

**Murat Manguoglu**
e-mail: mmanguog@cs.purdue.edu

**Ahmed H. Sameh**
e-mail: sameh@cs.purdue.edu

Department of Computer Science,
Purdue University,
305 North University Street,
West Lafayette, IN 47907

**Faisal Saied**
Computing Research Institute,
Purdue University,
Room 202,
250 North University Street,
West Lafayette, IN 47907
e-mail: fsaied@purdue.edu

**Tayfun E. Tezduyar**
e-mail: tezduyar@rice.edu

**Sunil Sathe**
e-mail: sathe@rice.edu

Mechanical Engineering,
Rice University,
MS 321,
6100 Main Street,
Houston, TX 77005

# Preconditioning Techniques for Nonsymmetric Linear Systems in the Computation of Incompressible Flows

*In this paper we present effective preconditioning techniques for solving the nonsymmetric systems that arise from the discretization of the Navier–Stokes equations. These linear systems are solved using either Krylov subspace methods or the Richardson scheme. We demonstrate the effectiveness of our techniques in handling time-accurate as well as steady-state solutions. We also compare our solvers with those published previously.*
[DOI: 10.1115/1.3059576]

## 1 Introduction

Increased emphasis on modeling of fluid-structure interaction (FSI) problems in recent years (see, for example, Refs. [1–36]) generated renewed interest in robust and efficient iterative solution techniques (see, for example, Refs. [12,13,30,37,38]) for the linear systems encountered in the computation of incompressible flows. These linear systems of equations can be written in the following general $2\times 2$ block form:

$$\begin{bmatrix} A & B \\ C^T & D \end{bmatrix}\begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ is nonsymmetric, $B, C \in \mathbb{R}^{n\times m}$, and $D \in \mathbb{R}^{m\times m}$, with $m < n$.

In this paper we study two model problems from incompressible flows: steady-state solution of lid-driven cavity flow and time-accurate solution of parachute problems.

The linear systems considered for the steady-state case are obtained from the IFISS [39] package for handling driven cavity problems (Oseen equations). The spatial discretization is based on the $Q_2/Q_1$ element, and the linear systems are derived after one nonlinear Picard iteration. This results in the following saddle-point problem:

$$\begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix}\begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \tag{2}$$

We note that solving Eq. (2) via a Krylov subspace method with a diagonal or incomplete $LU$ factorization preconditioner is either not possible or inefficient. An alternative approach is to use a block-diagonal or block $LU$ factorization based preconditioners. Such preconditioners require the solution of linear systems involving the Schur complement, $G = B^T A^{-1} B$, which is often expensive to form.

For systems in which $A$ is not far from being symmetric, Golub and Wathen [40] introduced the idea of using

$$M = \begin{bmatrix} A_s & B \\ B^T & 0 \end{bmatrix} \tag{3}$$

as a preconditioner, where $A_s = (A + A^T)/2$ is the symmetric part of $A$. Baggag and Sameh [37] extended this approach by introducing an inner Richardson iteration that gives excellent convergence rates for the inner-outer scheme, where the systems involving the Schur complement of the symmetric preconditioner are solved iteratively. A different approach for approximating the Schur complement matrix was introduced by Elman in Ref. [41]. In Elman's preconditioner, the action of $G^{-1}$ is approximated by

$$(B^T B)^{-1} B^T A B (B^T B)^{-1} \tag{4}$$

thus avoiding solving systems involving $G$.

The linear systems considered for the time-accurate solution cases are obtained from parachute aerodynamics computations [35] and are of the form given by Eq. (1). The linear systems

Contributed by the Applied Mechanics Division of ASME for publication in the JOURNAL OF APPLIED MECHANICS. Manuscript received February 26, 2008; final manuscript received August 14, 2008; published online January 14, 2009. Review conducted by Arif Masud.

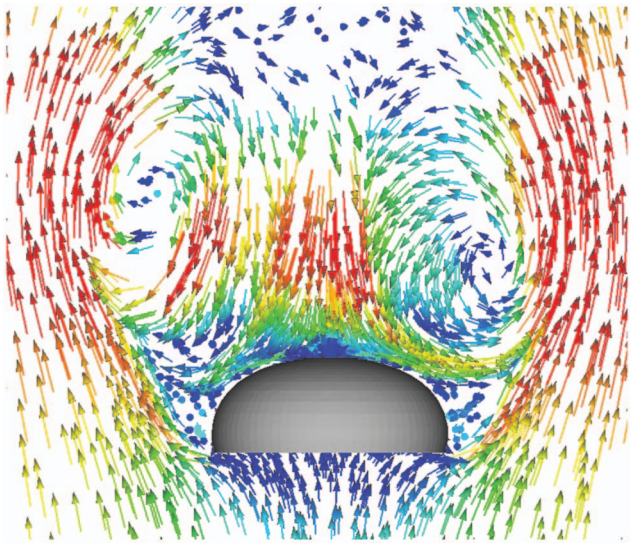


**Fig. 1 Flow around a parachute. The model and mesh are as given in Ref. [35]. The velocity vectors are colored by magnitude.**

tested were extracted from the first and fifth (last) nonlinear Newton–Rhapson iterations at a given time step of a computation with the fluid mechanics part of the SSTFSI-TIP1 technique (see Remarks 5 and 10 in Ref. [30] for this specific version of the stabilized space-time fluid-structure interaction (SSTFSI) technique). The nominal diameter of the parachute, which is held rigid during the computation, is about 120 ft (~36.6 m). The descent speed is 25 ft/s (~7.6 m/s), the air is assumed to have standard sea-level properties, and the Reynolds number is $1.8446\times10^7$. The time-step size is 0.116 s, and the flow is fully developed. Figure 1 shows the flow field.

## 2 Algorithms

Throughout this paper we use the term relative residual (or rel.res.) for $\|r_k\|_\infty/\|r_0\|_\infty$ where $r_k$ is the residual at the $k$th iteration.

### 2.1 Steady-State Case

*2.1.1 BFBT Preconditioner*. The BFB$^T$ (BFBT)preconditioner was introduced in Refs. [41,42] and later studied in Refs. [43–45]. It has the following block upper triangular form:

$$M=\begin{bmatrix} A & B \\ 0 & -\tilde{G} \end{bmatrix} \tag{5}$$

where $\tilde{G}$ is an approximation to the Schur complement matrix $G$. Let us assume that we would like to solve systems of the following form:

$$B^TA^{-1}Bz=t \tag{6}$$

Let

$$u=A^{-1}Bz \tag{7}$$

then

$$B^Tu=t \tag{8}$$

and the general solution of Eq. (8) can be written as

$$u=B(B^TB)^{-1}t+(I-B(B^TB)^{-1}B^T)v \tag{9}$$

where $v$ is arbitrary. Considering only the particular solution, we obtain

$$u=B(B^TB)^{-1}t \tag{10}$$

factor $A$ via MKL-PARDISO;
form $B^TB$;
factor $B^TB$ via MKL-PARDISO;
**solve** $\begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix}\begin{bmatrix} u \\ p \end{bmatrix}=\begin{bmatrix} f \\ g \end{bmatrix}$ via BiCGStab
  ($rel.res.\le\varepsilon_{out}$) with preconditioner $M=\begin{bmatrix} A & B \\ 0 & -\tilde{G} \end{bmatrix}$;
  where $\tilde{G}^{-1}=(B^TB)^{-1}B^TAB(B^TB)^{-1}$;
  **solve** $M\begin{bmatrix} v \\ w \end{bmatrix}=\begin{bmatrix} a \\ b \end{bmatrix}$
    **compute** $w=-(B^TB)^{-1}B^TAB(B^TB)^{-1}b$ (solve $(B^TB)x=y$ via MKL-PARDISO);
    **solve** $Av=a-Bw$ via MKL-PARDISO;
  **end**
**end**

**Fig. 2 BFBT preconditioner**

From Eqs. (7) and (10) we get

$$A^{-1}Bz=B(B^TB)^{-1}t \tag{11}$$

and

$$z=(B^TB)^{-1}B^TAB(B^TB)^{-1}t \tag{12}$$

Therefore, an approximation of $G^{-1}$ can be written as

$$\tilde{G}^{-1}=(B^TB)^{-1}B^TAB(B^TB)^{-1} \tag{13}$$

We note that the action of $\tilde{G}^{-1}$ on a vector $h$ can be realized by

$$(B^TB)^{-1}(B^TAB)(B^TB)^{-1}h \tag{14}$$

The BFBT algorithm is implemented as described in Fig. 2. The outer Krylov subspace method is BiCGStab [46], and the factorization of $A$ and $(B^TB)$ is done only once via MKL-PARDISO [47].

*2.1.2 Symmetric Nested Scheme*. Let $A_s=(A+A^T)/2$, and consider the following splitting of $A_s$ [48]:

$$A_s=R+Q \tag{15}$$

where $R$ contains the positive off-diagonal elements of $A_s$ and $Q$ contains the diagonal and negative off-diagonal elements of $A_s$. Let

$$\hat{A}_s=\hat{R}+Q \tag{16}$$

in which $\hat{R}$ is the diagonal matrix for which $\hat{R}e=Re$, where $e^T=[1,1,\ldots,1]$. $\hat{A}_s$ is thus a Stieltjes matrix. The nested scheme we propose consists of an outer BiCGStab iteration for solving Eq. (2), where the preconditioner given by Eq. (3) is solved by one step of a Richardson iteration. The preconditioner for the Richardson iteration is

$$\hat{M}=\begin{bmatrix} \hat{A}_s & B \\ B^T & 0 \end{bmatrix} \tag{17}$$

To solve systems involving $\hat{M}$,

$$\hat{M}\begin{bmatrix} v \\ w \end{bmatrix}=\begin{bmatrix} a \\ b \end{bmatrix} \tag{18}$$

consider the block $LDU$ factorization of $\hat{M}$,

$$\hat{M}=\begin{bmatrix} \hat{A}_s & 0 \\ B^T & I \end{bmatrix}\begin{bmatrix} \hat{A}_s^{-1} & 0 \\ 0 & -\hat{G} \end{bmatrix}\begin{bmatrix} \hat{A}_s & B \\ 0 & I \end{bmatrix} \tag{19}$$

Solving systems involving $\hat{A}_s$ is done directly via MKL-PARDISO, and solving systems involving the Schur complement $\hat{G}$ is done via the conjugate gradient (CG) method. In the CG scheme, matrix-vector products with $\hat{G}$ are achieved by first multiplying with $B$, solving a system involving $\hat{A}_s$ via MKL-PARDISO, followed

form $\hat{A}_s$ and factor it via MKL-PARDISO;
**solve** $\begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}$ via BiCGStab
  ($rel.res. \le \varepsilon_{out}$) with preconditioner $M = \begin{bmatrix} A_s & B \\ B & 0 \end{bmatrix}$;
  where $A_s = (A + A^T)/2$;
  **solve** $Mz = r$ via one step of Richardson
    $z_{k+1} = z_k + \hat{M}^{-1}(r - Mz_K)$ ;
    where $\hat{M} = \begin{bmatrix} \hat{A}_s & B \\ B^T & 0 \end{bmatrix}$;
    **solve** $\hat{M} \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$
      **solve** $\hat{A}_s v = a$ via MKL-PARDISO;
      **solve** $\hat{G} w = B^T v - b$ via CG with $rel.res. \le \varepsilon_{in}^{(1)}$
      (where $\hat{G} = B^T \hat{A}_s^{-1} B$);
      **solve** $\hat{A}_s v = a - Bw$ via MKL-PARDISO;
    **end**
  **end**
**end**

**Fig. 3 Symmetric nested preconditioner**

by a multiplication with $B^T$. We note that with this scheme we have three levels of nested iterations, as seen in Fig. 3.

*2.1.3 Nonsymmetric Nested Scheme*. The nonsymmetric nested preconditioner has outer BiCGStab iterations for solving Eq. (2), and the preconditioner is the coefficient matrix itself. The systems involving the preconditioner $M$,

$$M \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} \tag{20}$$

are solved by first forming the block *LDU* factorization

$$M = \begin{bmatrix} A & 0 \\ B^T & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & -G \end{bmatrix} \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \tag{21}$$

Solving Eq. (20) requires solving systems involving $A$ and $G$. Here, systems involving $A$ are solved via MKL-PARDISO, while those involving $G$ are solved via restarted GMRES(100) [49]. A description of this nonsymmetric scheme is given in Fig. 4.

**2.2 Time-Accurate Solutions.** Here, the linear systems are obtained from the first and fifth (last) nonlinear iterations of a parachute simulation. $A$ is of order 697,440 and $B$ has 121,370 columns. ILU(0) (zero fill-in incomplete LU factorization) and ILUT (dual threshold incomplete LU factorization) as preconditioners for BiCGStab for solving the system given by Eq. (1) have resulted in divergence. In what follows, we discuss two preconditioning strategies that proved to be quite effective in handling these linear systems.

*2.2.1 Reordering*. We reorder the above linear systems using the reverse Cuthill–McKee [50] scheme to obtain the symmetric permutations $P_A$ and $P_D$ for $A$ and $D$, respectively. Applying the

factor $A$ via MKL-PARDISO ;
**solve** $\begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}$ via BiCGStab
  ($rel.res. \le \varepsilon_{out}$) with preconditioner $M = \begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix}$;
  **solve** $M \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$
    **solve** $Av = a$ via MKL-PARDISO;
    **solve** $Gw = B^T v - b$ via GMRES(100) (with
    $rel.res. \le \varepsilon_{in}^{(2)}$, where $G = B^T A^{-1} B$);
    **solve** $Av = a - Bw$ via MKL-PARDISO;
  **end**
**end**

**Fig. 4 Nonsymmetric nested preconditioner**

compute $\hat{D} = D + \alpha I$;
compute ILU(0) factorization of $A \approx \tilde{A} = \tilde{L}_A \tilde{U}_A$;
compute ILU(0) factorization of $\hat{D} \approx \tilde{D} = \tilde{L}_{\hat{D}} \tilde{U}_{\hat{D}}$;
**solve** $\begin{bmatrix} A & B \\ C^T & D \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}$ via BiCGStab
  ($rel.res. \le \varepsilon_{out}$) with preconditioner $M = \begin{bmatrix} \tilde{A} & 0 \\ 0 & \tilde{D} \end{bmatrix}$;
  **solve** $M \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$
    **solve** $\tilde{A} v = a$ via triangular solves ;
    **solve** $\tilde{D} w = b$ via triangular solves ;
  **end**
**end**

**Fig. 5 Block-diagonal preconditioner**

permutations to Eq. (1),

$$\begin{bmatrix} P_A & 0 \\ 0 & P_D \end{bmatrix} \begin{bmatrix} A & B \\ C^T & D \end{bmatrix} \begin{bmatrix} P_A^T & 0 \\ 0 & P_D^T \end{bmatrix} \begin{bmatrix} \hat{u} \\ \hat{p} \end{bmatrix} = \begin{bmatrix} P_A & 0 \\ 0 & P_D \end{bmatrix} \begin{bmatrix} f \\ g \end{bmatrix} \tag{22}$$

yields the reordered system,

$$\begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C}^T & \hat{D} \end{bmatrix} \begin{bmatrix} \hat{u} \\ \hat{p} \end{bmatrix} = \begin{bmatrix} \hat{f} \\ \hat{g} \end{bmatrix} \tag{23}$$

This reordering needs to be done once since the sparsity structure of the coefficient matrix does not change during the nonlinear iterations. To simplify the notation, we assume that the system in Eq. (1) is the one resulting after the reordering.

*2.2.2 Block-Diagonal Preconditioner*. Since $D$ is symmetric positive semidefinite, $\hat{D} = D + \alpha I$ is nonsingular for a small positive $\alpha$. Clearly, $\alpha$ needs to be set sufficiently small so as not to introduce a large mass balance error. In our experiments, $\alpha$ is chosen to be $O(10^{-3})$ or less. Since the dimensions of the matrices $A$ and $D$ are too large for direct solvers, we use approximations (incomplete *LU* factorizations [51]) of these two matrices as preconditioners for outer BiCGStab iterations. The description of the algorithm is given in Fig. 5.

*2.2.3 Nested Preconditioner*. Similar to the block-diagonal preconditioner, let $\hat{D} = D + \alpha I$, and obtain the ILU(0) factorizations of $\hat{D}$ and $A$: $\tilde{A} = \tilde{L}_A \tilde{U}_A$ and $\tilde{D} = \tilde{L}_{\hat{D}} \tilde{U}_{\hat{D}}$. The outer Richardson iteration is given by

$$\begin{bmatrix} u_{k+1} \\ p_{k+1} \end{bmatrix} = \begin{bmatrix} u_k \\ p_k \end{bmatrix} + M^{-1} \left( \begin{bmatrix} f \\ g \end{bmatrix} - \begin{bmatrix} A & B \\ C^T & D \end{bmatrix} \begin{bmatrix} u_k \\ p_k \end{bmatrix} \right) \tag{24}$$

where

$$M = \begin{bmatrix} \tilde{A} & B \\ C^T & \tilde{D} \end{bmatrix} \tag{25}$$

Systems involving the preconditioner

$$M \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} \tag{26}$$

are solved by forming the block *LDU* factorization of $M$ as follows:

$$M = \begin{bmatrix} \tilde{A} & 0 \\ C^T & I \end{bmatrix} \begin{bmatrix} \tilde{A}^{-1} & 0 \\ 0 & -S \end{bmatrix} \begin{bmatrix} \tilde{A} & B \\ 0 & I \end{bmatrix} \tag{27}$$

where $S = \tilde{D} - C^T \tilde{A}^{-1} B$. Systems involving $\tilde{A}$ are solved directly via

compute $\hat{D} = D + \alpha I$;
compute ILU(0) factorization of $A \approx \tilde{A} = \tilde{L}_A \tilde{U}_A$;
compute ILU(0) factorization of $\hat{D} \approx \tilde{D} = \tilde{L}_{\hat{D}} \tilde{U}_{\hat{D}}$;
**solve** $\begin{bmatrix} A & B \\ C^T & D \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}$ via Richardson iterations
  ($rel.res \le \varepsilon_{out}$)
  $\begin{bmatrix} u_{k+1} \\ p_{k+1} \end{bmatrix} = \begin{bmatrix} u_k \\ p_k \end{bmatrix} + M^{-1}\left( \begin{bmatrix} f \\ g \end{bmatrix} - \begin{bmatrix} A & B \\ C^T & D \end{bmatrix} \begin{bmatrix} u_k \\ p_k \end{bmatrix} \right)$ where
  $M = \begin{bmatrix} \tilde{A} & B \\ C^T & \tilde{D} \end{bmatrix}$;
  **solve** $M \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix}$
    **solve** $\tilde{A}v = a$ via forward&backward subst.;
    **solve** $Sw = C^T v - b$ via BiCGStab ($rel.res. \le \varepsilon_{in}$) with preconditioner $\hat{M} = \tilde{D}$ (where $S = \tilde{D} - C^T \tilde{A}^{-1} B$);
    **solve** $\tilde{A}v = a - Bw$ via forward&backward subst.;
  **end**
**end**

**Fig. 6 Nested preconditioner**

forward and backward sweeps, while those involving the Schur complement $S$ are solved via BiCGStab with the preconditioner $\tilde{D}$. The description of the algorithm is given in Fig. 6. As in the block-diagonal preconditioner, there is an optimal choice of $\alpha$ as well as $\epsilon_{\text{in}}$. We refer the reader to Ref. [52] for a detailed discussion of determining the optimal paramaters in a similar problem.

## 3 Numerical Experiments

**3.1 Steady-State Case.** We implemented the algorithms described in Sec. 2.1 in FORTRAN90. All the computations are per-

**Table 1 Number of outer BiCGStab iterations for the BFBT scheme**

| Mesh | Re=10 | Re=50 | Re=100 | Re=500 |
|---|---|---|---|---|
| $32\times32$ | 17 | 20 | 25 | 52 |
| $64\times64$ | 25 | 33 | 38 | 86 |
| $128\times128$ | 41 | 48 | 62 | 87 |
| $256\times256$ | 62 | 84 | 100 | 171 |

**Table 2 (Average number of inner CG, number of outer BiCGStab) iterations for the symmetric nested scheme**

| Mesh | Re=10 | Re=50 |
|---|---|---|
| $32\times32$ | (5.6,4) | (4.4,13) |
| $64\times64$ | (5.8,5) | (3.8,11) |
| $128\times128$ | (6.1,5) | (3.7,10) |
| $256\times256$ | (6.3,4) | (3.6,9) |

**Table 3 Number of inner GMRES(100) iterations for the nonsymmetric nested scheme**

| Mesh | Re=10 | Re=50 | Re=100 | Re=500 |
|---|---|---|---|---|
| $32\times32$ | 33 | 56 | 82 | 374 |
| $64\times64$ | 34 | 52 | 76 | 464 |
| $128\times128$ | 33 | 52 | 77 | 386 |
| $256\times256$ | 33 | 53 | 78 | 439 |

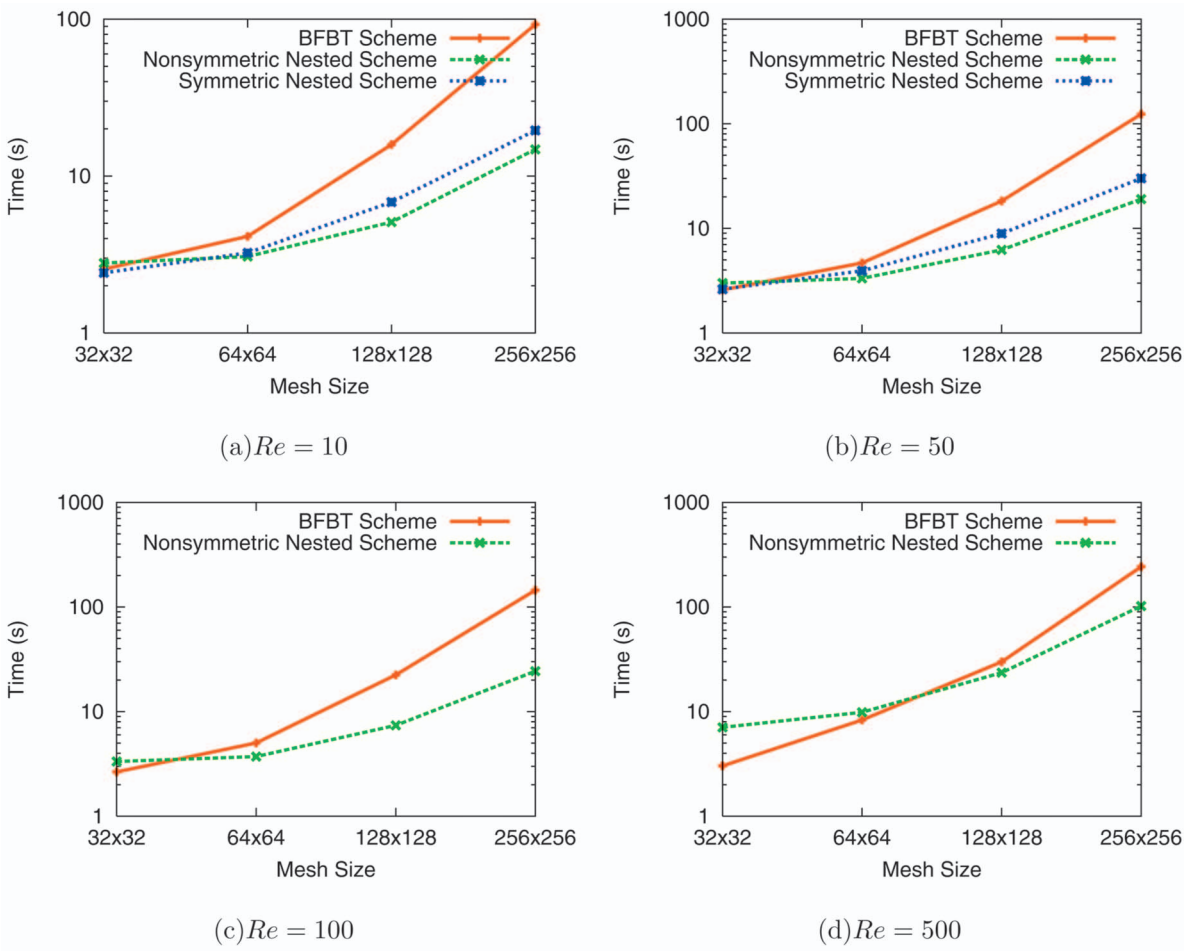



**Fig. 7 Total time for various mesh sizes and Reynold's numbers**

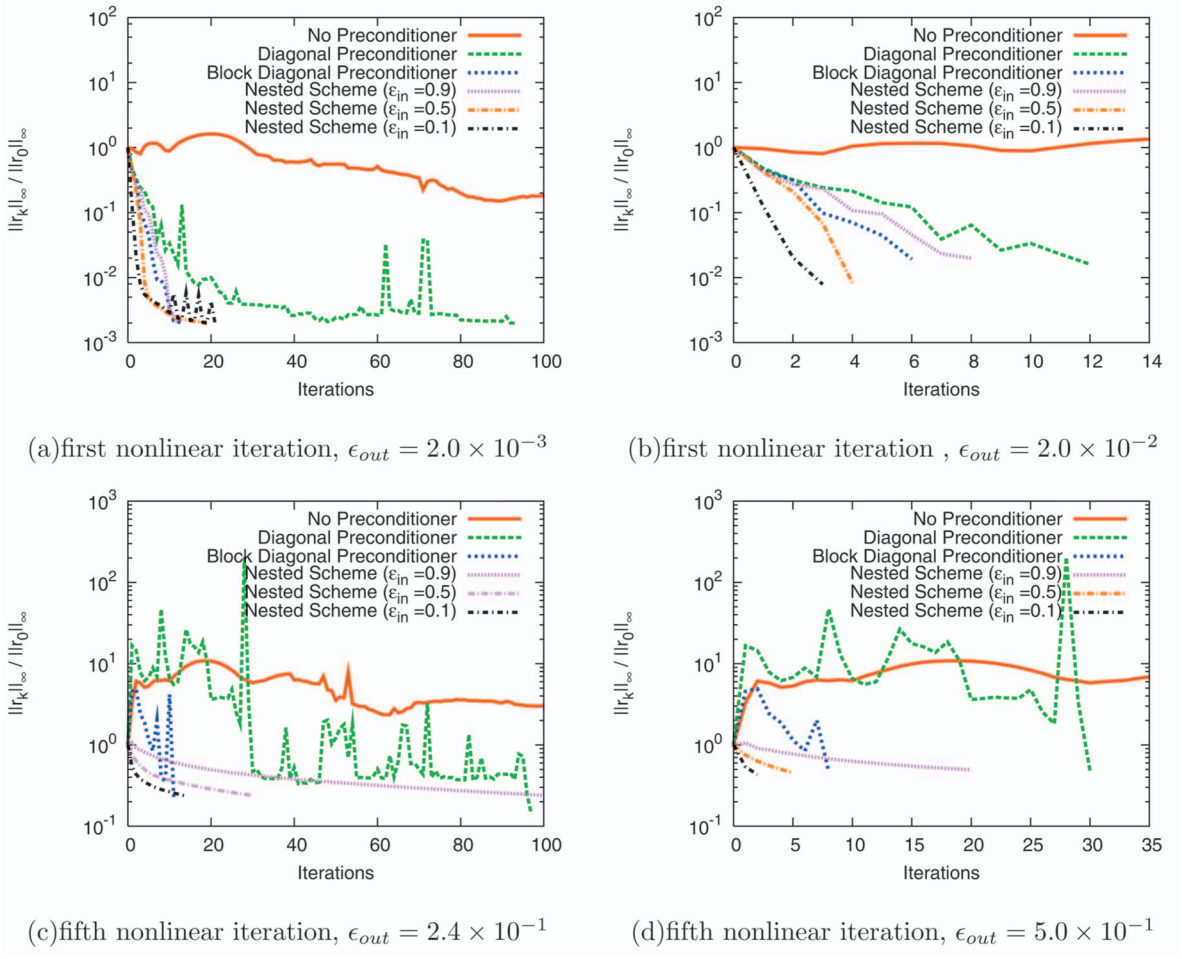



**Fig. 8 Relative residual plots**

formed on a single node of an Intel Xeon cluster. The outer stopping criterion for all three algorithms is $\epsilon_{\text{out}}=1\times10^{-5}$. For the symmetric nested scheme, the stopping criterion for the CG iterations is set to $\epsilon_{\text{in}}^{(1)}=1\times10^{-1}$. For the nonsymmetric nested scheme, the stopping criterion for GMRES is set to $\epsilon_{\text{in}}^{(2)}=1\times10^{-4}$.

We compare the performance of BFBT and our nonsymmetric nested schemes for those linear systems for which the Reynolds numbers are given by Re=10, 50, 100, and 500. For the symmetric nested scheme, we limit our tests to relatively small Reynolds numbers, 10 and 50. For all the tests, we use uniform meshes of sizes $32\times32$, $64\times64$, $128\times128$, and $256\times256$.

The number of iterations for the BFBT preconditioner is given in Table 1. In our experiments, the BFBT preconditioner shows dependence on both the mesh size and the Reynolds number. The number of inner CG iterations, as well as the outer BiCGStab iterations, in the symmetric nested scheme shows independence on the mesh size but weak dependence on the Reynolds number (see Table 2). With the nonsymmetric nested preconditioner, we require only 0.5 outer BiCGStab iterations. The number of inner GMRES iterations shows no dependence on the mesh size (see Table 3) except when Re=500. In this case, the restarted GMRES displays significant fluctuation in the number of iterations. For Re=500, even though the number of iterations is higher than that of the BFBT scheme, the total solution time is much smaller for the finer meshes, as shown in Fig. 7(*d*). The reason is that the cost of each iteration in the symmetric nested scheme is much lower than that in the BFBT scheme. Figures 7(*a*)–7(*d*) depict, for all three schemes, the total times for Re=10, 50, 100, and 500, respectively. For the coarsest mesh and the smallest Reynolds number, the symmetric nested scheme performs the best. As the Reynolds number becomes larger, the BFBT scheme is the fastest for coarser meshes. For fine meshes the nonsymmetric nested scheme outperforms the BFBT scheme significantly.

**3.2 Time-Accurate Solutions.** The initial residuals are $\|r_0\|_\infty=9.2\times10^{-5}$ and $\|r_0\|_\infty=2.0\times10^{-7}$ for the first and fifth nonlinear iterations, respectively. The iterations are terminated when $\|r_k\|_\infty/\|r_0\|_\infty\leq\epsilon_{\text{out}}$. Timings were done on a single processor of an Intel Quad-Core computing platform (Clovertwon).

For the first nonlinear iteration, we have experimented with two stopping criteria ($\epsilon_{\text{out}}$), $2.0\times10^{-3}$ and $2.0\times10^{-2}$, and three inner stopping criteria ($\epsilon_{\text{in}}$), $9.0\times10^{-1}$, $5.0\times10^{-1}$, and $1.0\times10^{-1}$. The outer relative residual plots for the first nonlinear iteration are given in Figs. 8(*a*) and 8(*b*).

For the fifth nonlinear iteration, the outer stopping criteria are $2.4\times10^{-1}$ and $5.0\times10^{-1}$, and the inner stopping criteria are $9.0\times10^{-1}$, $5.0\times10^{-1}$, and $1.0\times10^{-1}$. The outer relative residual plots for the fifth nonlinear iteration are given in Figs. 8(*c*) and 8(*d*).

The total times reported in Tables 4 and 5 include the reordering and factorization times (if any). For the first nonlinear iteration, when $\epsilon_{\text{out}}=2.0\times10^{-3}$ (see Table 4), the block-diagonal scheme converges in 13 iterations, consuming 39 s, almost three times faster than the diagonal preconditioned BiCGStab. The nested scheme converges in 51 s when $\epsilon_{\text{in}}=9.0\times10^{-1}$. If we aim, however, at a more relaxed stopping criterion ($\epsilon_{\text{out}}=2.0\times10^{-2}$, see Table 4), the diagonal preconditioned BiCGStab requires only 12 iterations consuming 13 s.

**Table 4 Timings for the first nonlinear iteration ($\epsilon_{out}=2.0\times10^{-3}$, $\epsilon_{out}=2.0\times10^{-2}$) and $\alpha=1.0\times10^{-5}$**

| Preconditioner | $\epsilon_{in}$ | Iterations: Inner (Avg.) | Iterations: Outer | Time (s) |
|---|---|---|---|---|
| None | - | - | (>100, >100) | (>102, >102) |
| Diagonal | - | - | (93,12) | (97,13) |
| Block-Diagonal | - | - | (13,6) | (39,23) |
| Nested | $9.0\times10^{-1}$ | (1.2,1.0) | (12,8) | (51,32) |
| Nested | $5.0\times10^{-1}$ | (1.3,1.0) | (19,4) | (80,23) |
| Nested | $1.0\times10^{-1}$ | (4.5,2.7) | (21,3) | (193,27) |

**Table 5 Timings for the fifth nonlinear iteration ($\epsilon_{out}=2.4\times10^{-1}$, $\epsilon_{out}=5.0\times10^{-1}$) and $\alpha=1.0\times10^{-3}$**

| Preconditioner | $\epsilon_{in}$ | Iterations: Inner (Avg.) | Iterations: Outer | Time (s) |
|---|---|---|---|---|
| None | - | - | (>100, >100) | (>101, <101) |
| Diagonal | - | - | (97,30) | (100,31) |
| Block-Diagonal | - | - | (11,8) | (34,27) |
| Nested | $9.0\times10^{-1}$ | (1.0,1.0) | (100,20) | (271,63) |
| Nested | $5.0\times10^{-1}$ | (1.0,1.0) | (30,5) | (112,27) |
| Nested | $1.0\times10^{-1}$ | (2.4,2.0) | (14,2) | (83,19) |

For the last nonlinear iteration, when $\epsilon_{out}=2.4\times10^{-1}$ (see Table 5), the block-diagonal preconditioned BiCGStab scheme converges in 11 iterations consuming 34 s. On the other hand, both the nested scheme and the diagonal preconditioned BiCGStab scheme consume more than 80 s. For a more relaxed stopping criterion, $\epsilon_{out}=5.0\times10^{-1}$ (see Table 5), the nested scheme consumes 19 s, while the block-diagonal and diagonal preconditioned BiCGStab schemes consume 27 s and 31 s, respectively.

## 4 Concluding Remarks

We have demonstrated the superiority of the nested algorithm for the steady-state cases that we considered in the computation of incompressible flows. For the time-accurate cases, we observe the following: (a) for the first nonlinear iteration, the diagonal preconditioned BiCGStab scheme is the best choice for relaxed stopping criteria, while the block-diagonal preconditioned BiCGStab is superior for tight stopping criteria; (b) for the fifth (last) nonlinear iteration, the block-diagonal preconditioned BiCGStab scheme is best suited for relaxed stopping criteria, while the nested scheme is the best for strict stopping criteria. There is no single algorithm that is best for all cases. The robustness of the nested scheme is clearly demonstrated in Fig. 8, exhibiting an almost monotone convergence of the residuals. Therefore, the nested scheme is the most robust method.

## Acknowledgment

This work has been partially supported by a grant from NSF (Grant No. NSF-CCF-0635169), a grant from DARPA/AFRL (Grant No. FA8750-06-1-0233), and a gift from Intel. The efforts of T.E.T. and S.S. were supported in part by NASA Johnson Space Center under Grant No. NNJ06HG84G and also in part by the Rice Computational Research Cluster funded by NSF under Grant No. CNS-0421109 and a partnership between Rice University, AMD, and Cray. We would like to thank Eric Polizzi for allowing us to use the Intel Clovertown Quad-Core computing platform.

## References

[1] Tezduyar, T., Aliabadi, S., Behr, M., Johnson, A., and Mittal, S., 1993, "Parallel Finite-Element Computation of 3D Flows," Computer, **26**(10), pp. 27–36.
[2] Tezduyar, T., Aliabadi, S., Behr, M., and Mittal, S., 1994, "Massively Parallel Finite Element Simulation of Compressible and Incompressible Flows," Comput. Methods Appl. Mech. Eng., **119**, pp. 157–177.
[3] Mittal, S., and Tezduyar, T., 1994, "Massively Parallel Finite Element Computation of Incompressible Flows Involving Fluid-Body Interactions," Comput. Methods Appl. Mech. Eng., **112**, pp. 253–282.
[4] Mittal, S., and Tezduyar, T. E., 1995, "Parallel Finite Element Simulation of 3D Incompressible Flows: Fluid-Structure Interactions," Int. J. Numer. Methods Fluids, **21**, pp. 933–953.
[5] Johnson, A., and Tezduyar, T., 1999, "Advanced Mesh Generation and Update Methods for 3D Flow Simulations," Comput. Mech., **23**, pp. 130–143.
[6] Kalro, V., and Tezduyar, T. E., 2000, "A Parallel 3D Computational Method for Fluid-Structure Interactions in Parachute Systems," Comput. Methods Appl. Mech. Eng., **190**, pp. 321–332.
[7] Stein, K., Benney, R., Kalro, V., Tezduyar, T. E., Leonard, J., and Accorsi, M., 2000, "Parachute Fluid-Structure Interactions: 3-D Computation," Comput. Methods Appl. Mech. Eng., **190**, pp. 373–386.
[8] Tezduyar, T., and Osawa, Y., 2001, "Fluid-Structure Interactions of a Parachute Crossing the Far Wake of an Aircraft," Comput. Methods Appl. Mech. Eng., **191**, pp. 717–726.
[9] Ohayon, R., 2001, "Reduced Symmetric Models for Modal Analysis of Internal Structural-Acoustic and Hydroelastic-Sloshing Systems," Comput. Methods Appl. Mech. Eng., **190**, pp. 3009–3019.
[10] Tezduyar, T., Sathe, S., Keedy, R., and Stein, K., 2004, "Space–time techniques for finite element computation of flows with moving boundaries and interfaces," Proceedings of the Third International Congress on Numerical Methods in Engineering and Applied Science, S. Gallegos, I. Herrera, S. Botello, F. Zarate, and G. Ayala, eds., Monterrey, Mexico, CD-ROM.
[11] Torii, R., Oshima, M., Kobayashi, T., Takagi, K., and Tezduyar, T., 2004, "Influence of Wall Elasticity on Image-Based Blood Flow Simulation," Trans. Jpn. Soc. Mech. Eng., Ser. A, **70**, pp. 1224–1231 (in Japanese).
[12] van Brummelen, E., and de Borst, R., 2005, "On the Nonnormality of Subiteration for a Fluid-Structure Interaction Problem," SIAM J. Sci. Comput. (USA), **27**, pp. 599–621.
[13] Michler, C., van Brummelen, E., and de Borst, R., 2005, "An Interface Newton–Krylov Solver for Fluid-Structure Interaction," Int. J. Numer. Methods Fluids, **47**, pp. 1189–1195.
[14] Gerbeau, J.-F., Vidrascu, M., and Frey, P., 2005, "Fluid-Structure Interaction in Blood Flow on Geometries Based on Medical Images," Comput. Struct., **83**, pp. 155–165.
[15] Tezduyar, T., Sathe, S., Keedy, R., and Stein, K., 2006, "Space-Time Finite Element Techniques for Computation of Fluid-Structure Interactions," Comput. Methods Appl. Mech. Eng., **195**, pp. 2002–2027.
[16] Tezduyar, T., Sathe, S., and Stein, K., 2006, "Solution Techniques for the Fully-Discretized Equations in Computation of Fluid-Structure Interactions With the Space-Time Formulations," Comput. Methods Appl. Mech. Eng., **195**, pp. 5743–5753.
[17] Torii, R., Oshima, M., Kobayashi, T., Takagi, K., and Tezduyar, T., 2006, "Computer Modeling of Cardiovascular Fluid-Structure Interactions With the Deforming-Spatial-Domain/Stabilized Space-Time Formulation," Comput. Methods Appl. Mech. Eng., **195**, pp. 1885–1895.
[18] Tezduyar, T., Sathe, S., Stein, K., and Aureli, L., 2006, "Modeling of Fluid-Structure Interactions With the Space-Time Techniques," *Fluid-Structure Interaction* (Lecture Notes in Computational Science and Engineering), H.-J. Bungartz and M. Schafer, eds., Springer, New York, Vol. 53, pp. 50–81.
[19] Torii, R., Oshima, M., Kobayashi, T., Takagi, K., and Tezduyar, T., 2006, "Fluid-Structure Interaction Modeling of Aneurysmal Conditions With High and Normal Blood Pressures," Comput. Mech., **38**, pp. 482–490.
[20] Dettmer, W., and Peric, D., 2006, "A Computational Framework for Fluid-Structure Interaction: Finite Element Formulation and Applications," Comput. Methods Appl. Mech. Eng., **195**, pp. 5754–5779.
[21] Bazilevs, Y., Calo, V., Huhes, T., and Zhang, Y., 2006, "Isogeometric Fluid-Structure Interaction Analysis With Applications to Arterial Blood Flow," Comput. Mech., **38**, pp. 310–322.
[22] Khurram, R., and Masud, A., 2006, "A Multiscale/Stabilized Formulation of the Incompressible Navier–Stokes Equations for Moving Boundary Flows and Fluid-Structure Interaction," Comput. Mech., **38**, pp. 403–416.
[23] Kuttler, U., Forster, C., and Wall, W., 2006, "A Solution for the Incompressibility Dilemma in Partitioned Fluid-Structure Interaction With Pure Dirichlet Fluid Domains," Comput. Mech., **38**, pp. 417–429.
[24] Lohner, R., Cebral, J., Yang, C., Baum, J., Mestreau, E. L., and Soto, O., 2006, "Extending the Range of Applicability of the Loose Coupling Approach for FSI Simulations," *Fluid-Structure Interaction* (Lecture Notes in Computational Science and Engineering), H.-J. Bungartz and M. Schafer, eds., Springer, New York, Vol. 53, pp. 82–100.
[25] Bletzinger, K.-U., Wuchner, R., and Kupzok, A., 2006, "Algorithmic Treatment of Shells and Free Form-Membranes in FSI," *Fluid-Structure Interaction* (Lecture Notes in Computational Science and Engineering), H.-J. Bungartz and M. Schafer, eds., Springer, New York, Vol. 53, pp. 336–355.
[26] Torii, R., Oshima, M., Kobayashi, T., Takagi, K., and Tezduyar, T., 2007, "Influence of Wall Elasticity in Patient-Specific Hemodynamic Simulations," Comput. Fluids, **36**, pp. 160–168.
[27] Masud, A., Bhanabhagvanwala, M., and Khurram, R., 2007, "An Adaptive Mesh Rezoning Scheme for Moving Boundary Flows and Fluid-Structure Interaction," Comput. Fluids, **36**, pp. 77–91.

[28] Sawada, T., and Hisada, T., 2007, "Fuid-Structure Interaction Analysis of the Two Dimensional Flag-in-Wind Problem by an Interface Tracking ALE Finite Element Method," Comput. Fluids, **36**, pp. 136–146.
[29] Wall, W., Genkinger, S., and Ramm, E., 2007, "A Strong Coupling Partitioned Approach for Fluid-Structure Interaction With Free Surfaces," Comput. Fluids, **36**, pp. 169–183.
[30] Tezduyar, T., and Sathe, S., 2007, "Modeling of Fluid-Structure Interactions With the Space-Time Finite Elements: Solution Techniques," Int. J. Numer. Methods Fluids, **54**, pp. 855–900.
[31] Tezduyar, T., Sathe, S., Cragin, T., Nanna, B., Conklin, B., Pausewang, J., and Schwaab, M., 2007, "Modeling of Fluid-Structure Interactions With the Space-Time Finite Elements: Arterial Fluid Mechanics," Int. J. Numer. Methods Fluids, **54**, pp. 901–922.
[32] Torii, R., Oshima, M., Kobayashi, T., Takagi, K., and Tezduyar, T., 2007, "Numerical Investigation of the Effect of Hypertensive Blood Pressure on Cerebral Aneurysm — Dependence of the Effect on the Aneurysm Shape," Int. J. Numer. Methods Fluids, **54**, pp. 995–1009.
[33] Tezduyar, T., Sathe, S., Schwaab, M., and Conklin, B., 2007, "Arterial Fluid Mechanics Modeling With the Stabilized Space-Time Fluid-Structure Interaction Technique," Int. J. Numer. Methods Fluids, **57**(5), pp. 601–629.
[34] Tezduyar, T., Sathe, S., Pausewang, J., Schwaab, M., Christopher, J., and Crabtree, J., 2008, "Interface Projection Techniques for Fluid–Structure Interaction Modeling With Moving-Mesh Methods," Comput. Mech., **43**(1), pp. 39–49.
[35] Tezduyar, T., Sathe, S., Pausewang, J., Schwaab, M., Christopher, J., and Crabtree, J., 2008, "Fluid-Structure Interaction Modeling of Ringsail Parachutes," Comput. Mech., **43**(1), pp. 133–142.
[36] Bazilevs, Y., Calo, V., Hughes, T., and Zhang, Y., 2008, "Isogeometric Fluid-Structure Interaction: Theory, Algorithms and Computations," unpublished.
[37] Baggag, A., and Sameh, A., 2004, "A Nested Iterative Scheme for Indefinite Linear Systems in Particulate Flows," Comput. Methods Appl. Mech. Eng., **193**, pp. 1923–1957.
[38] Tezduyar, T., and Sathe, S., 2005, "Enhanced-Discretization Successive Update Method (EDSUM)," Int. J. Numer. Methods Fluids, **47**, pp. 633–654.
[39] http://www.maths.manchester.ac.uk/djs/ifiss/
[40] Golub, G., and Wathen, A., 1998, "An Iteration for Indefinite Systems and Its Application to the Navier–Stokes Equations," SIAM J. Sci. Comput. (USA), **19**, pp. 530–539.
[41] Elman, H., 1999, "Preconditioning for the Steady-State Navier–Stokes Equations With Low Viscosity," SIAM J. Sci. Comput. (USA), **20**(4), pp. 1299–1316.
[42] Silvester, D., Elman, H., Kay, D., and Wathen, A., 2001, "Efficient Preconditioning of the Linearized Navier–Stokes Equations for Incompressible Flow," J. Comput. Appl. Math., **128**, pp. 261–279
[43] Elman, H., Silvester, D., and Wathen, A., 2005, *Finite Elements and Fast Iterative Solvers*, Oxford University Press, New York.
[44] Elman, H., Howle, V., Shadid, J., Shuttleworth, R., and Tuminaro, R., 2006, "Block Preconditioners Based on Approximate Commutators," SIAM J. Sci. Comput. (USA), **27**(5), pp. 1651–1668.
[45] Vainikko, E., and Graham, I., 2004, "A Parallel Solver for PDE Systems and Application to the Incompressible Navier–Stokes Equations," Appl. Numer. Math., **49**(1), pp. 97–116.
[46] van der Vorst, H., 1992, "BI-CGSTAB: A Fast and Smoothly Converging Variant of BI-CG for the Solution of Nonsymmetric Linear Systems," SIAM (Soc. Ind. Appl. Math.) J. Sci. Stat. Comput., **13**(2), pp. 631–644.
[47] Schenk, O., Gärtner, K., Fichtner, W., and Stricker, A., 2001, "PARDISO: A High-Performance Serial and Parallel Sparse Linear Solver in Semiconductor Device Simulation," FGCS, Future Gener. Comput. Syst., **18**(1), pp. 69–78.
[48] Axelsson, O., and Kolotilina, L., 2005, "Diagonally Compensated Reduction and Related Preconditioning Methods," Numer. Linear Algebra Appl., **1**(2), pp. 155–177.
[49] Saad, Y., and Schultz, M., 1986, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," SIAM (Soc. Ind. Appl. Math.) J. Sci. Stat. Comput., **7**, pp. 856–869.
[50] Cuthill, E., and McKee, J., 1969, "Reducing the Bandwidth of Sparse Symmetric Matrices," *Proceedings of the 1969 24th National Conference*, ACM, New York, NY, pp. 157–172.
[51] Saad, Y., 1990, "SPARSKIT: A Basic Tool Kit for Sparse Matrix Computations," NASA Ames Research Center, Technical Report No. 90-20.
[52] Manguoglu, M., Sameh, A. H., Tezduyar, T. E., and Sathe, S., 2008, "A Nested Iterative Scheme for Computation of Incompressible Flows in Long Domains," Comput. Mech., **43**(1), pp. 73–80.

**Ulrich Küttler**

**Wolfgang A. Wall**

Chair of Computational Mechanics,
TU München,
Boltzmannstrasse 15,
85747 Garching, Germany

# Vector Extrapolation for Strong Coupling Fluid-Structure Interaction Solvers

*Fluid-structure interaction (FSI) solvers based on vector extrapolation methods are discussed. The FSI solver framework builds on a Dirichlet–Neumann partitioning between general purpose fluid and structural solver. For strong coupling of the two fields vector extrapolation methods are employed to obtain a matrix free nonlinear solver. The emphasis of this presentation is on the embedding of well known vector extrapolation methods in a popular FSI solver framework and, in particular, the relation of these vector extrapolation methods to established fixed-point FSI schemes.* [DOI: 10.1115/1.3057468]

## 1 Introduction

The development of reliable fluid-structure interaction (FSI) solvers has seen major improvements in recent years. Along with ever increasing computational resources substantial theoretical work went into solver frameworks for various kinds of FSI applications. Among the many areas where FSI solvers are applied are aeroelasticity [1–3], civil engineering [4], and biomechanics [5–7]. As diverse as those areas of applications are, as variable are the FSI solution schemes that can be applied.

Crucial for all solution schemes of FSI problems is the general notion of coupling between the fluid and structural solver. These field solvers, however, are different in different applications. A variety of models have been applied on both the structural and the fluid side, so it is desirable to design flexible FSI schemes that can be used to couple different field solvers. And since weak coupling schemes are not applicable to some problem classes (see Ref. [8] for a respective analysis), such a solver should be able to realize a strong coupling. One FSI solver framework that enables such a flexibility is the strong coupling relaxation based fixed-point solver, which has a long tradition in FSI calculations, Refs. [9–12] among others.

Those fixed-point schemes, however, are nonlinear solvers for the interface degrees of freedom and are closely related to nonlinear vector extrapolation methods, which are well known in numerical mathematics, see Refs. [13–16]. These vector extrapolation schemes are based on the notion of a converging sequence of vectors, whose convergence can be accelerated by extrapolation. So, a generalization of the established fixed-point FSI solvers to solvers based on more history values promises alternative solution approaches and a better understanding of available methods. In this contribution the vector extrapolation implementation by Sidi [17] is applied to a FSI solver framework based on an incompressible Newtonian fluid and nonlinear elastodynamics, see Ref. [18].

Indeed, the generalization pursued here also sheds new light on some FSI solvers proposed in the literature, which have often been misinterpreted. In particular, it is shown that the Newton–Krylov FSI solver proposed by Michler et al. [19] should not be seen as a Newton based solver, but at its heart rather as a Krylov-based vector extrapolation scheme.

The remainder of this paper is organized as follows. In Sec. 2 field equations and coupling conditions of the actual FSI solver are sketched. Section 3 discusses alternative fixed-point relaxation methods that are generalized to polynomial based vector extrapolation schemes in Sec. 4. In Sec. 5 two FSI solvers that utilize vector extrapolation methods are discussed, and a numerical example is presented in Sec. 6.

## 2 Field Equations

FSI problems that are considered here are either two or three field problems. The structural field and the fluid field, the two physical fields, share a common interface. At this FSI interface $\Gamma$, coupling conditions

$$\mathbf{u}_\Gamma = \frac{\mathrm{d}\mathbf{d}_\Gamma}{\mathrm{d}t} \quad \text{and} \quad \boldsymbol{\sigma}_\Gamma^S \cdot \mathbf{n} = \boldsymbol{\sigma}_\Gamma^F \cdot \mathbf{n} \tag{1}$$

hold with the interface displacement $\mathbf{d}_\Gamma$, the interface velocity $\mathbf{u}_\Gamma$, the Cauchy stresses of both fluid $\boldsymbol{\sigma}_\Gamma^F$ and structure $\boldsymbol{\sigma}_\Gamma^S$, and the interface normal $\mathbf{n}$.

If an arbitrary Lagrangian–Eulerian approach (ALE) is employed, a third field, the nonphysical mesh field, is needed. This accounts for the fluid domain's deformation, an extension of the FSI interface deformation caused by the interaction. Thus a mapping is defined for the fluid domain displacements $\mathbf{d}^G$ based on the original fluid domain position $\mathbf{x}_0$ and the interface displacement $\mathbf{d}_\Gamma$

$$\mathbf{d}^G = \varphi(\mathbf{d}_\Gamma, \mathbf{x}_0, t) \quad \text{in} \quad \Omega^F \times (0,T) \tag{2}$$

The map (2) is arbitrary but unique. It should be noted that the same solution schemes, as proposed in this paper, can also be applied to fixed-grid FSI schemes [20,21].

The structural field is governed by the nonlinear elastodynamics equation

$$\rho^S \frac{\mathrm{d}^2\mathbf{d}}{\mathrm{d}t^2} = \boldsymbol{\nabla} \cdot (\mathbf{F} \cdot \mathbf{S}) + \rho^S \mathbf{b}^S \quad \text{in} \quad \Omega^S \times (0,T) \tag{3}$$

that determines the structural displacements $\mathbf{d}$ by prescribing an equilibrium between the body forces $\mathbf{b}^S$, the internal forces determined from the second Piola–Kirchhoff stress tensor $\mathbf{S}$ and the deformation gradient $\mathbf{F}$, and forces of inertia, where $\rho^S$ describes the structural density.

The fluid field is governed by the incompressible Navier–Stokes equations that read

$$\left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\mathbf{x}_0} + \mathbf{c} \cdot \boldsymbol{\nabla}\mathbf{u} - 2\nu\,\boldsymbol{\nabla} \cdot \boldsymbol{\epsilon}(\mathbf{u}) + \nabla p = \mathbf{b}^F \quad \text{in} \quad \Omega^F \times (0,T) \tag{4}$$

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0 \quad \text{in} \quad \Omega^F \times (0,T) \tag{5}$$

on a deforming domain, where both the fluid velocity $\mathbf{u}$ and the kinematic fluid pressure $p$ are unknown. The ALE convective ve-

Contributed by the Applied Mechanics Division of ASME for publication in the JOURNAL OF APPLIED MECHANICS. Manuscript received November 20, 2007; final manuscript received May 2, 2008; published online January 15, 2009. Review conducted by Tayfun E. Tezduyar.

locity $\mathbf{c}=\mathbf{u}-\mathbf{u}^G$ is the fluid velocity relative to the arbitrarily moving fluid domain with the domain velocity determined by mapping (2)

$$\mathbf{u}^G = \left.\frac{\partial \varphi}{\partial t}\right|_{\mathbf{x}_0} \tag{6}$$

Equation (4) states the conservation of momentum, where $\mathbf{b}^F$ represents the fluid body forces, $\boldsymbol{\epsilon}(\mathbf{u})$ the strain rate tensor of the Newtonian fluid, and $\nu$ the kinematic viscosity. Equation (5) states the conservation of mass, which requires the fluid's incompressibility due to a constant density $\rho^F$.

The field equations come with the appropriate initial and boundary conditions. These conditions are important but rather unspectacular and will not be discussed in detail. See Ref. [18] for a more complete presentation of the FSI solver framework.

Ultimately, the field equations are discretized using finite elements in space. In time, implicit finite difference schemes are used. This results in a set of nonlinear algebraic equations that has to be solved for each time step

$$\mathbf{M}(\mathbf{d}^{G,n+1},\mathbf{d}_\Gamma^{n+1}) = 0 \tag{7}$$

$$\mathbf{S}(\mathbf{d}^{n+1}) = \mathbf{f}^{S,n+1} \tag{8}$$

$$\mathbf{F}(\mathbf{u}^{n+1},p^{n+1},\mathbf{d}^{G,n+1}) = \mathbf{f}^{F,n+1} \tag{9}$$

where a concrete mapping (2) has to be chosen for a meaningful discretization (7). The discrete structural solver (8) and the discrete fluid solver (9) are straight versions of Eqs. (3) and (4) combined with Eq. (5), respectively.

Equations (7)–(9) are already prepared to become the building blocks of a Dirichlet–Neumann partitioned fixed-point FSI solver. The coupling conditions that these equations have to satisfy are the coupling of structural displacements and fluid velocities at the interface

$$\mathbf{u}_\Gamma^{n+1} = \frac{\mathbf{d}_\Gamma^{n+1}-\mathbf{d}_\Gamma^{n}}{\Delta t} \tag{10}$$

the equilibrium at the interface

$$\mathbf{f}_\Gamma^{F,n+1} = -\mathbf{f}_\Gamma^{S,n+1} \tag{11}$$

and, of course, the fluid domain deformation according to the interface movement

$$\mathbf{d}_\Gamma^{G,n+1} = \mathbf{d}_\Gamma^{n+1} \tag{12}$$

## 3 Relaxation Accelerated Fixed-Point Schemes

**3.1 Fixed-Point Fluid-Structure Interaction Coupling.** Fixed-point solvers are very well established for FSI problems. For a recent presentation, see Ref. [18]. The general idea of fixed-point solvers is to calculate a new interface displacement $\tilde{\mathbf{d}}_{\Gamma,i+1}^{n+1}$ out of the current one, $\mathbf{d}_{\Gamma,i}^{n+1}$, by cycling through field solvers (7)–(9). The subscript $i$ specifies the iteration count.

$$\tilde{\mathbf{d}}_{\Gamma,i+1}^{n+1} = \mathbf{S}_\Gamma^{-1}(\mathbf{F}_\Gamma(\mathbf{d}_{\Gamma,i}^{n+1})) \tag{13}$$

The interface operators $\mathbf{S}_\Gamma^{-1}$ and $\mathbf{F}_\Gamma$ in Eq. (13) contain the solution of the structural field (8) and the fluid field (9) together with the fluid mesh movement (7), respectively. The fluid interface operator $\mathbf{f}_{\Gamma,i}^{F,n+1}=\mathbf{F}_\Gamma(\mathbf{d}_{\Gamma,i}^{n+1})$ therefore abbreviates an algorithm that (a) prescribes the interface displacement $\mathbf{d}_{\Gamma,i}^{n+1}$ to mesh equation (7) according to Eq. (12), (b) solves mesh equation (7), (c) prescribes the interface velocity $\mathbf{u}_{\Gamma,i}^{n+1}$ from condition (10) to fluid equation (9), (d) solves fluid equation (9) on the deforming fluid domain, and (e) extracts the interface tractions $\mathbf{f}_{\Gamma,i}^{F,n+1}$ from the fluid field.

In comparison the structural interface operator $\tilde{\mathbf{d}}_{\Gamma,i+1}^{n+1}=\mathbf{S}_\Gamma^{-1}(\mathbf{f}_{\Gamma,i}^{n+1})$ simply describes the process that (a) loads structural equation (8) with the interface load $\mathbf{f}_{\Gamma,i}^{F,n+1}$, (b) solves structural equation (8), and (c) returns new interface displacements $\tilde{\mathbf{d}}_{\Gamma,i+1}^{n+1}$.

Thus each FSI cycle (13) requires solving all three field equations (7)–(9). The interface equilibrium (11) is satisfied by design, the coupling condition (10) however might be violated by the new interface displacement $\tilde{\mathbf{d}}_{\Gamma,i+1}^{n+1}$. So after each FSI cycle (13), the interface residual

$$\mathbf{r}_{\Gamma,i+1}^{n+1} = \tilde{\mathbf{d}}_{\Gamma,i+1}^{n+1} - \mathbf{d}_{\Gamma,i}^{n+1} \tag{14}$$

provides a measure of the solution quality. If the convergence criterion

$$\frac{1}{\sqrt{n_{\text{eq}}}}|\mathbf{r}_{\Gamma,i+1}^{n+1}| < \epsilon_\Gamma \tag{15}$$

with specified $\epsilon_\Gamma$ is met, the latest interface displacement of this iterative scheme $\tilde{\mathbf{d}}_{\Gamma,i+1}^{n+1}$ is taken for the time step's solution. Otherwise the interface displacement is relaxed

$$\mathbf{d}_{\Gamma,i+1}^{n+1} = \mathbf{d}_{\Gamma,i}^{n+1} + \omega_i \mathbf{r}_{\Gamma,i+1}^{n+1} = \omega_i \tilde{\mathbf{d}}_{\Gamma,i+1}^{n+1} + (1-\omega_i)\mathbf{d}_{\Gamma,i}^{n+1} \tag{16}$$

with a potential iteration specific relaxation factor $\omega_i$, and the calculation proceeds with a new FSI cycle (13).

The relaxation of the interface displacements (16) is crucial in this algorithm to enforce and accelerate convergence. This is a direct outcome of the Dirichlet–Neumann partitioning, which generally overestimates the structural stiffness for the fluid solver, and accordingly overestimates the fluid forces that load the structure as well. All but the most forgiving FSI problems will diverge if no relaxation (i.e., $\omega_i=1$) is applied.

In this framework the crucial ingredient is thus the calculation of $\omega_i$. Hence, methods for calculating $\omega_i$ are given as follows.

**3.2 Aitken's $\Delta^2$ Method.** A method that has been very successfully used for FSI calculations, see Refs. [10,4,18], and others, is Aitken's $\Delta^2$ method, as suggested by Irons and Tuck [22]. This method starts with two known pairs of interface displacements $(\tilde{\mathbf{d}}_{\Gamma,i},\mathbf{d}_{\Gamma,i-1})$ and $(\tilde{\mathbf{d}}_{\Gamma,i+1},\mathbf{d}_{\Gamma,i})$ and finds the relaxation parameter

$$\omega_i = -\omega_{i-1}\frac{(\mathbf{r}_{\Gamma,i})^T(\mathbf{r}_{\Gamma,i+1}-\mathbf{r}_{\Gamma,i})}{|\mathbf{r}_{\Gamma,i+1}-\mathbf{r}_{\Gamma,i}|^2} \tag{17}$$

to calculate $\mathbf{d}_{\Gamma,i+1}$ from $\tilde{\mathbf{d}}_{\Gamma,i+1}$ and $\mathbf{d}_{\Gamma,i}$ through Eq. (16). The idea behind this formula is the scalar secant method, which is given by

$$d_{\Gamma,i+1} = \frac{d_{\Gamma,i-1}\tilde{d}_{\Gamma,i+1}-\tilde{d}_{\Gamma,i}d_{\Gamma,i}}{d_{\Gamma,i-1}-\tilde{d}_{\Gamma,i}-d_{\Gamma,i}+\tilde{d}_{\Gamma,i+1}} \tag{18}$$

$$= \frac{d_{\Gamma,i-1}\tilde{d}_{\Gamma,i+1}-\tilde{d}_{\Gamma,i}d_{\Gamma,i}}{r_{\Gamma,i+1}-r_{\Gamma,i}} \tag{19}$$

In the vector case the division by $r_{\Gamma,i+1}-r_{\Gamma,i}$ is replaced by the vector inverse $(\mathbf{r}_{\Gamma,i+1}-\mathbf{r}_{\Gamma,i})/|\mathbf{r}_{\Gamma,i+1}-\mathbf{r}_{\Gamma,i}|^2$. See Ref. [18] for more details on this relaxation method.

**3.3 Alternative Aitken's $\Delta^2$ Methods.** There are more vector versions of Aitken's $\Delta^2$ method available, see, for instance, Ref. [23]. Most of these, however, are not based on pairs of vectors, but on a sequence of three vectors $\mathbf{d}_{\Gamma,i}^{n+1}$, $\mathbf{d}_{\Gamma,i+1}^{n+1}$, and $\mathbf{d}_{\Gamma,i+2}^{n+1}$. Such a sequence can be generated from a known interface displacement $\mathbf{d}_{\Gamma,i}^{n+1}$ by repeated evaluation of the FSI cycle (13) and the relaxation step (16). The relaxation (16) is required at this point to keep the successive FSI evaluations (13) from diverging. Thus a suitable problem dependent relaxation factor

$$\omega_i = \text{const} \tag{20}$$

needs to be chosen in Eq. (16). This way a prerelaxed sequence of three interface displacement vectors $\mathbf{d}_{\Gamma,i}^{n+1}$, $\mathbf{d}_{\Gamma,i+1}^{n+1}$, and $\mathbf{d}_{\Gamma,i+2}^{n+1}$ can be generated with the prerelaxed residual

$$\bar{\mathbf{r}}_{\Gamma,i+1}^{n+1} = \mathbf{d}_{\Gamma,i+1}^{n+1} - \mathbf{d}_{\Gamma,i}^{n+1} \tag{21}$$

"Prerelaxed" thereby means relaxation with a chosen constant relaxation parameter before the actual relaxation scheme—here the alternative Aitken $\Delta^2$ method—starts working. This notation will also be used in Secs. 4–6. The aim of the Aitken $\Delta^2$ methods, like those in Ref. [23], is to find improved approximations of the interface displacement

$$\hat{\mathbf{d}}_{\Gamma,i+1}^{n+1} = \bar{\omega}\mathbf{d}_{\Gamma,i+1}^{n+1} + (1-\bar{\omega})\mathbf{d}_{\Gamma,i}^{n+1} \tag{22}$$

and

$$\hat{\mathbf{d}}_{\Gamma,i+2}^{n+1} = \bar{\omega}\mathbf{d}_{\Gamma,i+2}^{n+1} + (1-\bar{\omega})\mathbf{d}_{\Gamma,i+1}^{n+1} \tag{23}$$

that are closer to solution $\mathbf{d}_{\Gamma}^{n+1}$ than the three known interface displacements. Therefore, the relaxed interface residual

$$\hat{\mathbf{r}}_{\Gamma,i+2}^{n+1} = \hat{\mathbf{d}}_{\Gamma,i+2}^{n+1} - \hat{\mathbf{d}}_{\Gamma,i+1}^{n+1} \tag{24}$$

should be minimal

$$|\hat{\mathbf{r}}_{\Gamma,i+2}^{n+1}|^2 \rightarrow \min \tag{25}$$

This condition determines the relaxation factor

$$\bar{\omega} = -\frac{(\bar{\mathbf{r}}_{\Gamma,i+1})^T(\bar{\mathbf{r}}_{\Gamma,i+2} - \bar{\mathbf{r}}_{\Gamma,i+1})}{|\bar{\mathbf{r}}_{\Gamma,i+2} - \bar{\mathbf{r}}_{\Gamma,i+1}|^2} \tag{26}$$

to be used in Eq. (23) to get the relaxed interface displacement $\hat{\mathbf{d}}_{\Gamma,i+2}^{n+1}$.

There is a remarkable similarity between relaxation parameter definitions (17) and (26). The only structural difference is the recursion in Eq. (17) that is absent in Eq. (26). However, whereas Eq. (17) is based on the direct interface residual $\mathbf{r}_{\Gamma,i}$ and can be calculated after each FSI cycle, relaxation version (26) is built on the prerelaxed residual $\bar{\mathbf{r}}_{\Gamma,i}$ and needs two previous FSI cycles before it can be applied. For this reason the Aitken $\Delta^2$ version (17) proposed by Irons and Tuck [22] requires about half the number of FSI cycles compared with other versions such as those considered by MacLeod [23], e.g., Eq. (26). This is especially important to note since several authors in the past used Eq. (26) instead of Eq. (17) and referred to this approach as the Aitken relaxation with a reference to Ref. [10], see, e.g., Refs. [24,25].

However, it is possible to improve the relaxation methods that are based on minimizing the relaxed residual (25) by incorporating more than three history values, see Ref. [26]. These methods are known as vector extrapolation methods.

**3.4 Vector Extrapolation.** Vector extrapolation methods find approximate solutions based on the first members of a converging vector series. A linear combination of $k$ members $\mathbf{d}_{\Gamma,i+1}^{n+1}, \mathbf{d}_{\Gamma,i+2}^{n+1}, \ldots, \mathbf{d}_{\Gamma,i+k}^{n+1}$ of a vector series of interface displacements that converges to the solution $\mathbf{d}_{\Gamma}^{n+1}$ yields the approximation

$$\hat{\mathbf{d}}_{\Gamma,i+k}^{n+1} = \mathbf{d}_{\Gamma,i+1}^{n+1} + \sum_{j=2}^{k} \bar{\omega}_j(\mathbf{d}_{\Gamma,i+j}^{n+1} - \mathbf{d}_{\Gamma,i+j-1}^{n+1}) \tag{27}$$

with unknown extrapolation factors $\bar{\omega}_j$. As stated above, FSI cycle (13) can be used to generate such a sequence from a known displacement $\mathbf{d}_{\Gamma,i}^{n+1}$, if a suitable fixed relaxation parameter $\omega_i$ is applied in the associated relaxation steps (16). Through this converging vector series a new sequence of $k$ difference vectors $\bar{\mathbf{r}}_{\Gamma,i+j}^{n+1}$ is generated. And since the vector series converges, the difference between two adjacent members, the prerelaxed residual of one FSI cycle (13), tends to zero

$$\lim_{j\rightarrow\infty}(\mathbf{d}_{\Gamma,i+j}^{n+1} - \mathbf{d}_{\Gamma,i+j-1}^{n+1}) = \lim_{j\rightarrow\infty} \bar{\mathbf{r}}_{\Gamma,i+j}^{n+1} \rightarrow 0 \tag{28}$$

The same extrapolation as above can be applied to the residual vectors

$$\hat{\mathbf{r}}_{\Gamma,i+k}^{n+1} = \bar{\mathbf{r}}_{\Gamma,i+1}^{n+1} + \sum_{j=2}^{k} \bar{\omega}_j(\bar{\mathbf{r}}_{\Gamma,i+j}^{n+1} - \bar{\mathbf{r}}_{\Gamma,i+j-1}^{n+1}) \tag{29}$$

$$= \bar{\mathbf{r}}_{\Gamma,i+1}^{n+1} + \sum_{j=2}^{k} \bar{\omega}_j \Delta\bar{\mathbf{r}}_{\Gamma,i+j}^{n+1} \tag{30}$$

with the difference

$$\Delta\bar{\mathbf{r}}_{\Gamma,i+j}^{n+1} = \bar{\mathbf{r}}_{\Gamma,i+j}^{n+1} - \bar{\mathbf{r}}_{\Gamma,i+j-1}^{n+1} \tag{31}$$

Since the limit of series (28) is zero, the minimization of extrapolated residual (29)

$$|\hat{\mathbf{r}}_{\Gamma,i+k}^{n+1}|^2 \rightarrow \min \tag{32}$$

can be perceived as a least-squares approach to find the extrapolation factors $\bar{\omega}_j$. With these factors the extrapolation of the original sequence (27) can be pursued.

For a sequence of length $k=2$, the minimization of Eq. (32) leads to relaxation factor (26).

However, as the iteration proceeds and $\bar{\mathbf{r}}_{\Gamma,i+j}^{n+1} \rightarrow 0$ the difference (31) will lead to severe cancellations, and the least-squares fit will fail for purely numerical reasons. So it is advisable to avoid the differences in Eq. (29) and to simply rewrite the equation to extrapolate the prerelaxed residuals instead

$$\hat{\mathbf{r}}_{\Gamma,i+k}^{n+1} = \bar{\mathbf{r}}_{\Gamma,i+1}^{n+1} + \sum_{j=2}^{k} \bar{\omega}_j(\bar{\mathbf{r}}_{\Gamma,i+j}^{n+1} - \bar{\mathbf{r}}_{\Gamma,i+j-1}^{n+1}) = \sum_{j=1}^{k} \bar{\gamma}_j \bar{\mathbf{r}}_{\Gamma,i+j}^{n+1} \tag{33}$$

The extrapolation factors $\bar{\gamma}_j$ are

$$\bar{\gamma}_j = \bar{\omega}_j - \bar{\omega}_{j+1} \tag{34}$$

with $\bar{\omega}_1 = 1$ and $\bar{\omega}_{k+1} = 0$. A least-squares approach to determine factor $\bar{\gamma}_j$ from Eq. (33) leads to a homogeneous linear system. Factor $\bar{\gamma}_j$ needs to be normalized

$$\gamma_j = \bar{\gamma}_j / \sum_{i=1}^{k} \bar{\gamma}_i \tag{35}$$

in order to obtain a valid extrapolation of the interface displacements

$$\hat{\mathbf{d}}_{\Gamma,i+k}^{n+1} = \sum_{j=1}^{k} \gamma_j \mathbf{d}_{\Gamma,i+j}^{n+1} \tag{36}$$

The extrapolated interface displacement $\hat{\mathbf{d}}_{\Gamma,i+k}^{n+1}$ can then become the starting point of a new sequence of displacement vectors

$$\mathbf{d}_{\Gamma,i+k+1}^{n+1} = \hat{\mathbf{d}}_{\Gamma,i+k}^{n+1} \tag{37}$$

that can be extrapolated again.

The method shown here is a vector extrapolation method known as reduced rank extrapolation (RRE), see, e.g., Refs. [13,14].

## 4 Vector Extrapolation Framework

There is a variety of vector extrapolation methods, where the major two categories are polynomial methods, see Refs. [13,14], and methods based on the $\epsilon$-algorithm, as shown by Brezinski and Zaglia [27] and Brezinski [15]. In this presentation only the first category is considered.

The three main polynomial vector extrapolation methods, see Ref. [16], can be neatly arranged in a common framework. In the context of FSI solvers, starting with a sequence of $k+1$ known interface displacements $\mathbf{d}_{\Gamma,i}^{n+1}, \mathbf{d}_{\Gamma,i+1}^{n+1}, \mathbf{d}_{\Gamma,i+2}^{n+1}, \ldots, \mathbf{d}_{\Gamma,i+k}^{n+1}$, the extrapolation can be stated as demonstrated above

$$\hat{\mathbf{d}}_{\Gamma,i+k}^{n+1} = \sum_{j=1}^{k} \gamma_j \mathbf{d}_{\Gamma,i+j}^{n+1} \tag{38}$$

where $\gamma_j$ is determined by

$$\sum_{j=1}^{k} \alpha_{l,j} \gamma_j = 0 \tag{39}$$

under the constraint

$$\sum_{j=1}^{k} \gamma_j = 1 \tag{40}$$

Different extrapolation methods are obtained for different choices of $\alpha_{l,j}$. Popular methods are as follows:

- minimal polynomial extrapolation (MPE)

$$\alpha_{l,j} = \bar{\mathbf{r}}_{\Gamma,i+l}^{n+1} \cdot \bar{\mathbf{r}}_{\Gamma,i+j}^{n+1} \tag{41}$$

- reduced rank extrapolation

$$\alpha_{l,j} = \Delta\bar{\mathbf{r}}_{\Gamma,i+l}^{n+1} \cdot \bar{\mathbf{r}}_{\Gamma,i+j}^{n+1} \tag{42}$$

- modified minimal polynomial extrapolation (MMPE)

$$\alpha_{l,j} = \mathbf{y}_{\Gamma,l}^{n+1} \cdot \bar{\mathbf{r}}_{\Gamma,i+j}^{n+1} \tag{43}$$

where $\mathbf{y}_{\Gamma,l}^{n+1}$ is a set of linear independent vectors.

In all cases an overdetermined linear system of equations with unknown extrapolation factor $\gamma_j$ is obtained. The methods differ in how the extrapolation factors are calculated to minimize the residual (33). Both MPE and RRE require the solution of a linear least-squares system to determine the extrapolation factors $\gamma_j$.

**4.1 Krylov Space Based Implementation.** If the operators in Eq. (13) are linear, the generated prerelaxed sequence of interface displacements $\mathbf{d}_{\Gamma,i+j}^{n+1}$ clearly builds a Krylov space. Vector extrapolation (38) finds an approximation within that Krylov space that minimizes the residual. Thus, it is tempting to exploit the Krylov space for a practical implementation of vector extrapolation methods. This has been done by Sidi [17] for both MPE and RRE. Indeed, as MPE and RRE applied to a linear operator are equivalent to Arnoldi's method and GMRES, respectively, the implementation proposed by Sidi [17] follows the same lines as the implementation of GMRES by Saad and Schultz [28]. An implementation for MMPE has been proposed by Jbilou and Sadok [29]. In the present contribution the vector extrapolation algorithm by Sidi [17] is applied to the Dirichlet–Neumann coupled FSI problems.

The vector extrapolation algorithm proposed by Sidi [17] works with a successively enlarging sequence of vectors. Each time a new interface displacement $\mathbf{d}_{\Gamma,i+j}^{n+1}$ is added to the sequence, the extrapolation of residual (33) is calculated. The algorithm is constructed such that the linear least-squares system is successively built and factorized. Each iteration of the extrapolation algorithm consists of the following three steps.

1. Enlarge and factorize the linear least-squares problem using the modified Gram–Schmidt process.
2. Calculate extrapolation factors $\gamma_j$ according to the chosen vector extrapolation method.
3. Extrapolate residual $\hat{\mathbf{r}}_{\Gamma,i+k}^{n+1}$ using Eq. (33) and extrapolate displacement $\hat{\mathbf{d}}_{\Gamma,i+k}^{n+1}$ using Eq. (36).

Normally the extrapolated interface displacement $\hat{\mathbf{d}}_{\Gamma,i+k}^{n+1}$ will not be the solution, but closer to the solution than any of the original displacements in the sequence. However, to find out how good the extrapolated displacements actually are, another FSI cycle (13) is needed. In order to avoid that and still be able to assess the quality of the extrapolation, the extrapolated interface residual $\hat{\mathbf{r}}_{\Gamma,i+k}^{n+1}$ can be used, which comes out of the algorithm for free. But this is just extrapolated as well, so it cannot be trusted too much.

In general the above algorithm will be used in cycles. That is, a sequence of up to $k$ interface displacements $\mathbf{d}_{\Gamma,i+k}^{n+1}$ is calculated, each one requiring a full FSI cycle. After each FSI cycle, the set of interface residuals is extrapolated. If the extrapolated residual is small enough, the extrapolation of the interface displacements is done, otherwise the next FSI cycle is calculated.

Once the extrapolation is finished the real interface residual $\mathbf{r}_{\Gamma,i+k+1}^{n+1}$ is built and tested using Eqs. (13)–(15)

$$\mathbf{r}_{\Gamma,i+k+1}^{n+1} = \mathbf{S}_{\Gamma}^{-1}(\mathbf{F}_{\Gamma}(\hat{\mathbf{d}}_{\Gamma,i+k}^{n+1})) - \hat{\mathbf{d}}_{\Gamma,i+k}^{n+1} \tag{44}$$

and if convergence has not yet been achieved, the extrapolated interface displacement $\hat{\mathbf{d}}_{\Gamma,i+k}^{n+1}$ becomes the start of a new sequence to be extrapolated again (Eq. (37)). This cycle is executed until the final residual (44) satisfies the tolerance (15).

## 5 Alternative Vector Extrapolation Based FSI Solvers

**5.1 Krylov-Based Vector Extrapolation Solver.** Newton–Krylov solvers (see Ref. [30]) have successfully been applied to Dirichlet–Neumann coupled FSI problems by Gerbeau and Vidrascu [24], Gerbeau et al. [31], Fernández and Moubachir [32], and others.

Another Dirichlet–Neumann partitioned FSI scheme has been published by Michler et al. [19] and was also named interface Newton–Krylov solver. An error-amplification analysis for this scheme is provided in Ref. [33]. In our opinion the scheme proposed by Michler et al. [19] however should not be named a Newton–Krylov solver since it does not utilize the interface Jacobian

$$\mathbf{J}_{\Gamma} = \frac{\partial \mathbf{r}_{\Gamma}}{\partial \mathbf{d}_{\Gamma}} \tag{45}$$

in any way. Instead a least-squares problem built from interface residual differences (29) is solved. So, from our point of view the scheme is a Krylov-based vector extrapolation method.

*Remark.* We firmly believe that a proper and specific nomenclature is an invaluable ingredient in a scientific discussion about different methods. For this reason we follow Knoll and Keyes [30] and restrict the name "Jacobian-free Newton–Krylov" to methods that utilize a Jacobian-vector product to link a nonlinear Newton iteration with a Krylov-based iterative linear solver. The term "Jacobian-free" is used to denote the absence of an explicitly constructed Jacobian matrix in the solution process of Eq. (45). If linear system (45) is avoided altogether, the resulting coupling scheme is not of Newton-type. Accordingly the label Newton–Krylov is inappropriate for the coupling method by Michler et al. [19].

In its basic form, the algorithm proposed by Michler et al. [19] constitutes the RRE method built on the unfavorable numerically sensitive residual extrapolation (29). The numerical sensitivities are relaxed to some extent by residual difference definition (31) relative to the first residual in the extrapolation step

$$\Delta\bar{\mathbf{r}}_{\Gamma,i+j}^{n+1} = \bar{\mathbf{r}}_{\Gamma,i+j}^{n+1} - \bar{\mathbf{r}}_{\Gamma,i+1}^{n+1} \tag{46}$$

The relaxation (16) with a fixed parameter $\omega_i$, which is needed to obtain a converging sequence of interface displacements $\mathbf{d}_{\Gamma,i+j}^{n+1}$ in the first place, is not explicitly mentioned, but easy to add. The proposed residual extrapolation can be exchanged for the more robust version (33). Thus a slightly modified version of the basic algorithm by Michler et al. [19] is obtained and should constitute a reasonable solution approach for FSI problems.

A little disturbance is caused by the proposed Gram–Schmidt orthonormalization of interface residual $\bar{\mathbf{r}}_{\Gamma,i+j}^{n+1}$ though. This orthonormalization leads to trial interface displacement $\mathbf{d}_{\Gamma,i+j}^{n+1}$ that are very different from the solution $\mathbf{d}_{\Gamma}^{n+1}$, and thus are incredibly

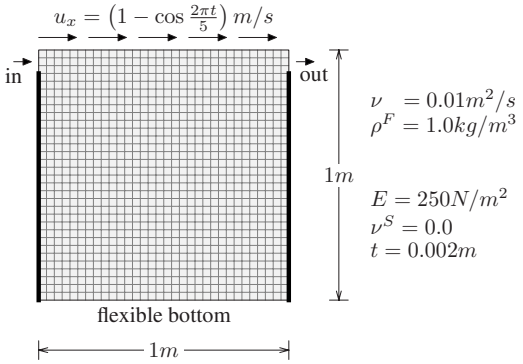



**Fig. 1 Driven cavity with flexible bottom**

hard for the nonlinear FSI field solvers (Eq. (13)). In Ref. [19] the authors augmented the orthonormalization step by a user supplied relaxation constant $\nu$ to cure these kinds of problems. However, it turns out that the suggested algorithm is extremely sensitive with respect to the chosen parameter $\nu$. Indeed the best choice of $\nu$, the one that undoes the damage done by the orthonormalization, changes with each iteration and cannot be supplied by the user. Hence, it is advisable to drop the orthonormalization altogether.

The Gram–Schmidt process introduced in Sec. 4.1 and the one suggested by Michler et al. [19] have little in common. The former is part of the solution process of the least-squares problem, which is intrinsic to vector extrapolation methods. The latter changes the creation of the vector sequence without any effect on the least-squares problem itself. Indeed, the residual differences (Eq. (46)) used by Michler et al. [19] are not orthogonal to each other, since the orthonormalization is done before the FSI field solvers (Eq. (13)) are called.

The second modification proposed by Michler et al. [19] concerns the reuse of the generated Krylov space for further extrapolation steps. This idea is certainly compelling and should be pursued further.

**5.2 Field Solver Approximation by Vector Extrapolation.** An interesting idea of how to apply vector extrapolation for FSI solvers has been suggested by Vierendeels [34]. Here the vector extrapolation scheme is used to predict the fluid's interface force $\mathbf{f}_{\Gamma,i+j}^{n+1}$ (or rather just the interface pressure) based on previous fluid solver results and currently prescribed interface displacement $\mathbf{d}_{\Gamma,i+j}^{n+1}$. Afterwards a monolithic coupling of the structural solver and the extrapolation scheme for the fluid solver, the reduced-order model, is pursued, such that the structural solver incorporates a rough estimate of the fluid solver's sensitivities. These solvers are used in Ref. [34] to build a coupling scheme without relaxation of interface displacements (Eq. (16)).

An extension of this idea is suggested as well. There, both fluid and structural solver predictors are based on vector extrapolation. In this case the two extrapolation schemes are coupled to assess the influence of one field on the other. The actual field solver both remain untouched, thus black box solvers can be used. See Ref. [34] for details on these coupling schemes and some numerical results.

## 6 Numerical Example

**6.1 Driven Cavity With Flexible Bottom.** The example used to demonstrate the vector extrapolation methods is a simple 2D driven cavity with flexible bottom. This example has been introduced in Ref. [35] and has been used for a variety of numerical studies since then. The cavity has the shape of a unit square and a flexible bottom (Fig. 1). At the top a time-dependent horizontal velocity is prescribed. The fluid domain is discretized with stabilized Q1Q1 elements in a uniform $32 \times 32$ element mesh. The cavity is defined as a leaky cavity where at both sides two unconstrained nodes allow free in- and outflow of fluid (Fig. 1). This way the structural displacements are not constrained by the fluid's incompressibility.

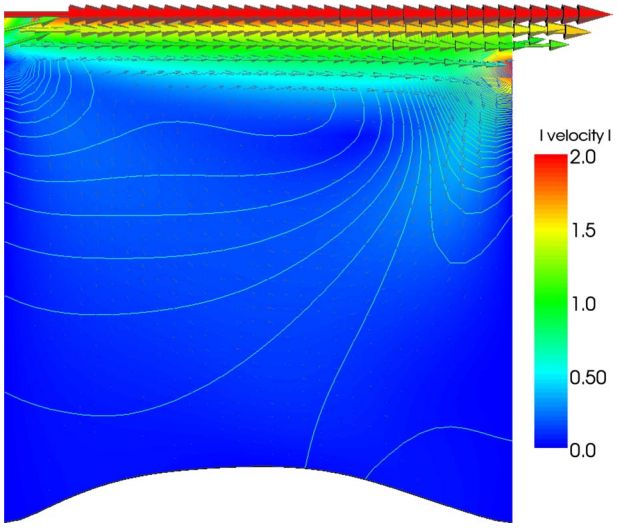



**Fig. 2 Driven cavity velocity and pressure solution at $t$ = 7.5 s**

The simplicity of the example (and the very coarse mesh used here) keeps the calculation time low, so many test calculations can be run. Furthermore, the example is constructed such that a variation in the structural density $\rho^S$ yields various variants that pose very different challenges to FSI coupling algorithms [8,18].

In Fig. 2 the velocity and pressure solution for the case with a structural density $\rho^S = 500\ \mathrm{kg/m^3}$ at time $t = 7.5$ s is shown. The high structural density $\rho^S = 500\ \mathrm{kg/m^3}$ leads to coupled system (13), which is solved within just a few coupling iterations. Figure 3 shows the number of FSI iterations for each time step for the Aitken method with relaxation factor (17) and for the reduced rank and minimal polynomial vector extrapolation methods.

The first relaxation factor in each time step for the Aitken methods has been constrained by setting $\omega_i^{n+1} = \max(\omega_k^n, 0.1)$. The basic fixed relaxation used to get a vector sequence to extrapolate, i.e., the prerelaxed series, has been chosen as $\omega_i = 0.005$. The maximum allowed vector sequence length has been set to $k = 10$ for both vector extrapolation methods. Furthermore, the tolerance required in order to exit a vector extrapolation and to start a new sequence is

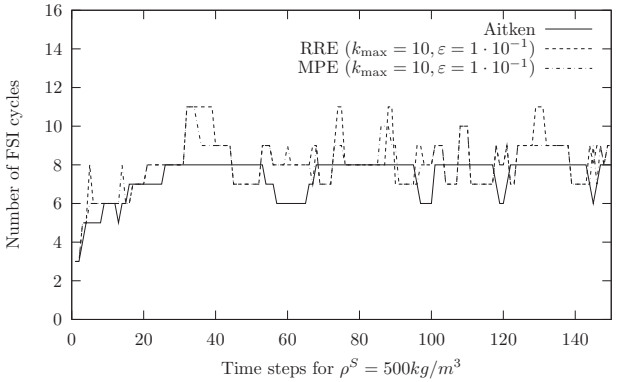



**Fig. 3 Number of FSI cycles for driven cavity with structural density $\rho^S = 500\ \mathrm{kg/m^3}$ and $k_{\max} = 10$**

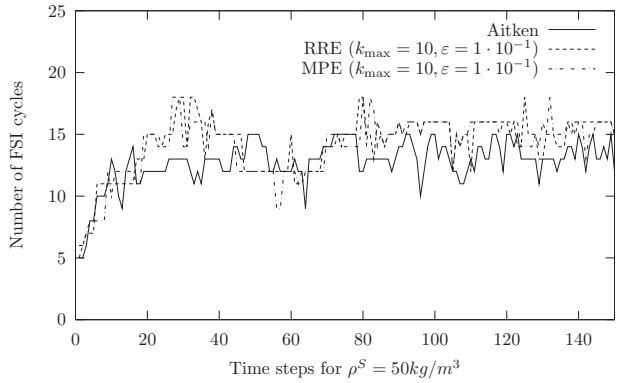



**Fig. 4 Number of FSI cycles for driven cavity with structural density $\rho^S$=50 kg/m$^3$ and $k_{\max}$=10**

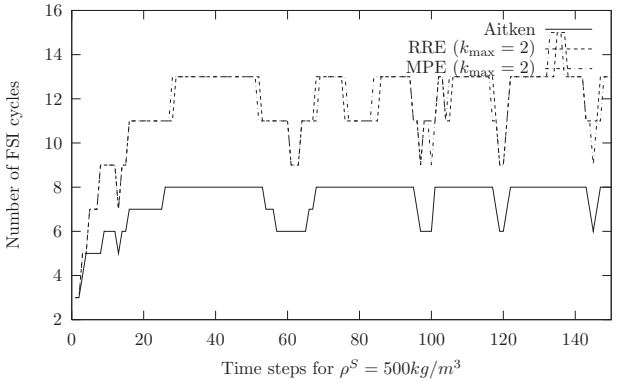



**Fig. 6 Number of FSI cycles for driven cavity with structural density $\rho^S$=500 kg/m$^3$ and $k_{\max}$=2**

$$\frac{|\mathbf{r}_{\Gamma,i+k}^{n+1}|}{|\mathbf{r}_{\Gamma,i}^{n+1}|} \leq \epsilon = 1 \times 10^{-1} \tag{47}$$

for the extrapolated interface residual (33). The reason for such a high tolerance is the nonlinear nature of FSI coupling. A lower tolerance would lead to a much better linear approximation of the interface displacements, however the nonlinear FSI coupling iteration would not gain much. A tolerance of $\epsilon = 1 \times 10^{-2}$ already increases the number of FSI cycles per time step considerably. So due to the high tolerance (47), on average there are just three FSI iterations needed for both extrapolation methods, which corresponds to a vector extrapolation sequence of length 4.

As can be seen in Fig. 3, the vector extrapolation methods require a few more FSI cycles than the Aitken relaxation method. And because the evaluation of the FSI field solvers dominates the calculation time, the additional work required by the Krylov vector extrapolation solver from Ref. [17] can be neglected, the Aitken relaxation method is the fastest solver in this case.

This picture does not change much in Fig. 4, where the FSI iteration counts obtained for a structural density of $\rho^S$ =50 kg/m$^3$ are shown. The number of FSI cycles needed increased for all methods, and the Aitken relaxation method is the fastest one again. The average number of FSI cycles needed to reach the tolerance (47) in an extrapolation step is 3.7 (RRE) and 3.8 (MPE).

With a structural density of $\rho^S$=5 kg/m$^3$ the picture finally changes, see Fig. 5. Here the vector extrapolation methods require less FSI cycles than the Aitken method, with an average of 4.6 and 4.7 FSI cycles per extrapolation for RRE and MPE, respectively.

Constraining the allowed length of vector sequences to $k$=2, which corresponds to the alternative Aitken version with the relaxation parameter (26), leading to a very different behavior. In this case the vector extrapolation methods require far more FSI cycles than the Aitken method for $\rho^S$=500 kg/m$^3$ (see Fig. 6).

Decreasing the structural density only increases the gap between these methods, as can be seen in Figs. 7 and 8. In particular for $\rho^S$=5 kg/m$^3$ MPE repeatedly hits the maximum allowed number of 100 vector extrapolations (with two FSI cycle evaluations each) within one time step and goes on without a fully converged interface displacement (15). So the Aitken relaxation definition (17) is indeed a much better choice than the definition (26).

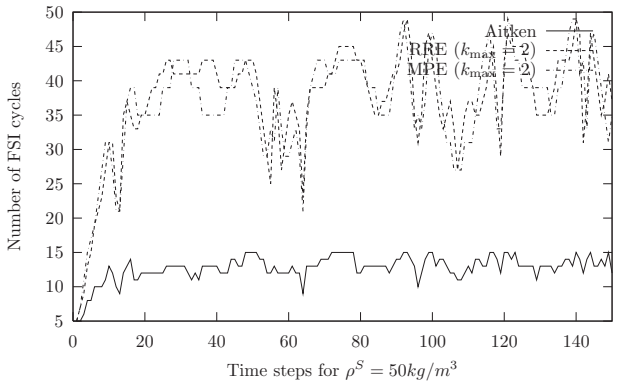



**Fig. 7 Number of FSI cycles for driven cavity with structural density $\rho^S$=50 kg/m$^3$ and $k_{\max}$=2**

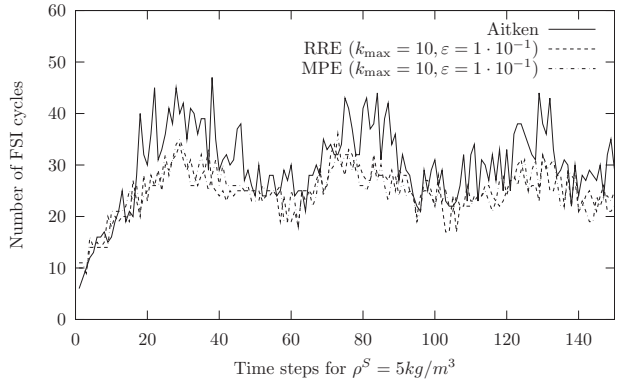



**Fig. 5 Number of FSI cycles for driven cavity with structural density $\rho^S$=5 kg/m$^3$ and $k_{\max}$=10**

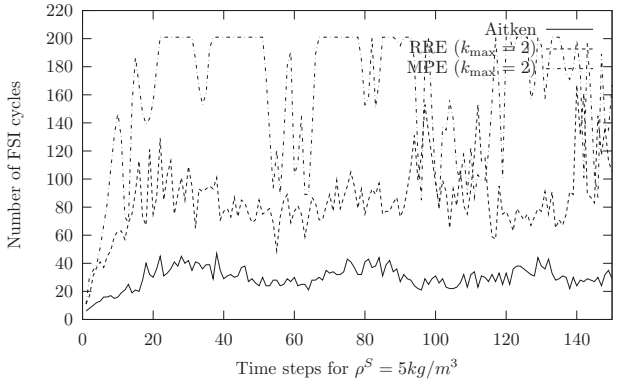



**Fig. 8 Number of FSI cycles for driven cavity with structural density $\rho^S$=5 kg/m$^3$ and $k_{\max}$=2**

## 7 Conclusion

The nonlinear system of equations that describes a Dirichlet–Neumann partitioned FSI problem has been solved using vector extrapolation methods. To do this the FSI solver framework presented in Ref. [18] has been extended based on a vector extrapolation implementation idea, as given in Ref. [17]. The relation of vector extrapolation methods to fixed-point FSI solvers based on Aitken relaxation is discussed in detail. Other FSI solvers suggested in literature are found to be customized versions of vector extrapolation methods. A numerical example has been discussed in detail to show the performance of vector extrapolation in comparison to the fixed-point method with Aitken relaxation. It is shown that Aitken relaxation is much simpler to implement and yields a faster coupling scheme in many cases. Thus the Aitken scheme seems to be a proper choice for many FSI problems.

## References


[1] Kalro, V., and Tezduyar, T. E., 2001, "A Parallel 3D Computational Method for Fluid-Structure Interactions in Parachute Systems," Comput. Methods Appl. Mech. Eng., **190**, pp. 321–332.

[2] Stein, K., Tezduyar, T., and Benney, R., 2003, "Computational Methods for Modeling Parachute Systems," Comput. Sci. Eng., **5**(1), pp. 39–46.

[3] Farhat, C., 2004, "CFD-Based Nonlinear Computational Aeroelasticity," *Encyclopedia of Computational Mechanics*, Vol. 3, E. Stein, R. D. Borst, and T. Hughes, eds., Wiley, New York, Chap. 13.

[4] Wüchner, R., Kupzok, A., and Bletzinger, K.-U., 2007, "A Framework for Stabilized Partitioned Analysis of Thin Membrane-Wind Interaction," Int. J. Numer. Methods Fluids, **54**(6–8), pp. 945–963.

[5] Bazilevs, Y., Calo, V. M., Zhang, Y., and Hughes, T. J. R., 2006, "Isogeometric Fluid-Structure Interaction Analysis With Applications to Arterial Blood Flow," Comput. Mech., **38**(4–5), pp. 310–322.

[6] Tezduyar, T. E., and Sathe, S., 2007, "Modeling of Fluid-Structure Interactions With the Space-Time Finite Elements: Solution Techniques," Int. J. Numer. Methods Fluids, **54**, pp. 855–900.

[7] Tezduyar, T. E., Sathe, S., Cragin, T., Nanna, B., Conklin, B. S., Pausewang, J., and Schwaab, M., 2007, "Modelling of Fluid-Structure Interactions With the Space-Time Finite Elements: Arterial Fluid Mechanics," Int. J. Numer. Methods Fluids, **54**(6–8), pp. 901–922.

[8] Förster, C., Wall, W. A., and Ramm, E., 2007, "Artificial Added Mass Instabilities in Sequential Staggered Coupling of Nonlinear Structures and Incompressible Viscous Flows," Comput. Methods Appl. Mech. Eng., **196**, pp. 1278–1293.

[9] Wall, W. A., Mok, D. P., and Ramm, E., 1999, "Partitioned Analysis Approach of the Transient Coupled Response of Viscous Fluids and Flexible Structures," *Solids, Structures and Coupled Problems in Engineering*, Proceedings of ECCM '99, Munich, Germany, W. Wunderlich, ed.

[10] Mok, D. P., and Wall, W. A., 2001, "Partitioned Analysis Schemes for the Transient Interaction of Incompressible Flows and Nonlinear Flexible Structures," *Trends in Computational Structural Mechanics*, W. A. Wall, K.-U. Bletzinger, and K. Schweitzerhof, eds., CIMNE, Barcelona.

[11] Le Tallec, P., and Mouro, J., 2001, "Fluid Structure Interaction With Large Structural Displacements," Comput. Methods Appl. Mech. Eng., **190**(24–25), pp. 3039–3067.

[12] Tezduyar, T. E., Sathe, S., Keedy, R., and Stein, K., 2006, "Space-Time Finite Element Techniques for Computation of Fluid-Structure Interactions," Comput. Methods Appl. Mech. Eng., **195**, pp. 2002–2027.

[13] Smith, D. A., Ford, W. F., and Sidi, A., 1987, "Extrapolation Methods for Vector Sequences," SIAM Rev., **29**(2), pp. 199–233.

[14] Smith, D. A., Ford, W. F., and Sidi, A., 1988, "Erratum: Extrapolation Methods for Vector Sequences," SIAM Rev., **30**(4), pp. 623–624.

[15] Brezinski, C., 2000, "Convergence Acceleration During the 20th Century," J. Comput. Appl. Math., **122**, pp. 1–21.

[16] Jbilou, K., and Sadok, H., 2000, "Vector Extrapolation Methods, Applications and Numerical Comparison," J. Comput. Appl. Math., **122**, pp. 149–165.

[17] Sidi, A., 1991, "Efficient Implementation of Minimal Polynomial and Reduced Rank Extrapolation Methods," J. Comput. Appl. Math., **36**(3), pp. 305–337.

[18] Küttler, U., and Wall, W. A., 2008, "Fixed-Point Fluid-Structure Interaction Solvers With Dynamic Relaxation," Comput. Mech., **43**(1), pp. 61–72.

[19] Michler, C., van Brummelen, E. H., and de Borst, R., 2005, "An Interface Newton–Krylov Solver for Fluid-Structure Interaction," Int. J. Numer. Methods Fluids, **47**, pp. 1189–1195.

[20] Gerstenberger, A., and Wall, W. A., 2008, "An Extended Finite Element Method/Lagrange Multiplier Based Approach for Fluid-Structure Interaction," Comput. Methods Appl. Mech. Eng., **197**, pp. 1699–1714.

[21] Wall, W. A., Gamnitzer, P., and Gerstenberger, A., 2008, "Fluid-Structure Interaction Approaches on Fixed Grids Based on Two Different Domain Decomposition Ideas," Int. J. Comput. Fluid Dyn., **22**(6), pp. 411–427.

[22] Irons, B., and Tuck, R. C., 1969, "A Version of the Aitken Accelerator for Computer Implementation," Int. J. Numer. Methods Eng., **1**, pp. 275–277.

[23] MacLeod, A. J., 1986, "Acceleration of Vector Sequences by Multi-Dimensional $\delta^2$ Methods," Commun. Appl. Numer. Methods, **2**, pp. 385–392.

[24] Gerbeau, J.-F., and Vidrascu, M., 2003, "A Quasi-Newton Algorithm Based on a Reduced Model for Fluid-Structure Interaction Problems in Blood Flows," Math. Modell. Numer. Anal., **37**(4), pp. 631–647.

[25] Calvo, F. J., Margetts, L., Gabaldón, F., and Romero, I., 2007, "Parallel Three Dimensional Analysis of a Lid Driven Cavity Coupled to a Flexible Moving Base," Proceedings of Coupled Problems, Ibiza, Spain, pp. 623–626.

[26] Hafez, M., Parlette, E., and Salas, M., 1986, "Convergence Acceleration of Iterative Solutions of Euler Equations for Transonic Flow Computations," Comput. Mech., **1**(3), pp. 165–176.

[27] Brezinski, C., and Zaglia, M. R., 1991, *Extrapolation Methods: Theory and Practice* (Studies in Computational Mathematics 2), North-Holland, Amsterdam.

[28] Saad, Y., and Schultz, M. H., 1986, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," SIAM (Soc. Ind. Appl. Math.) J. Sci. Stat. Comput., **7**(3), pp. 856–869.

[29] Jbilou, K., and Sadok, H., 1999, "LU Implementation of the Modified Minimal Polynomial Extrapolation Method for Solving Linear and Nonlinear Systems," IMA J. Numer. Anal., **19**(4), pp. 549–561.

[30] Knoll, D. A., and Keyes, D. E., 2004, "Jacobian-Free Newton–Krylov Methods: A Survey of Approaches and Applications," J. Comput. Phys., **193**, pp. 357–397.

[31] Gerbeau, J.-F., Vidrascu, M., and Frey, P., 2005, "Fluid-Structure Interaction in Blood Flows on Geometries Coming From Medical Imaging," Comput. Struct., **83**, pp. 155–165.

[32] Fernández, M., and Moubachir, M., 2005, "A Newton Method Using Exact Jacobians for Solving Fluid-Structure Coupling," Comput. Struct., **83**(2–3), pp. 127–142.

[33] Michler, C., van Brummelen, E. H., and de Borst, R., 2006, "Error-Amplification Analysis of Subiteration-Preconditioned GMRES for Fluid-Structure Interaction," Comput. Methods Appl. Mech. Eng., **195**, pp. 2124–2148.

[34] Vierendeels, J., 2006, "Implicit Coupling of Partitioned Fluid-Structure Interaction Solvers Using Reduced-Order Models," *Fluid-Structure Interaction: Modelling, Simulation, Optimisation* (LNCSE), Vol. 53, Springer, New York, pp. 1–18.

[35] Wall, W. A., 1999, "Fluid-Struktur-Interaktion mit Stabilisierten Finiten Elementen," Ph.D. thesis, Institut für Baustatik, Universität Stuttgart, Stuttgart.

# Added Mass Effects of Compressible and Incompressible Flows in Fluid-Structure Interaction

**E. H. van Brummelen**
Faculty of Mechanical, Maritime and Materials Engineering,
Delft University of Technology,
Mekelweg 2,
Delft, 2628 CD, Netherlands
e-mail: e.h.vanbrummelen@tudelft.nl

*The subiteration method, which forms the basic iterative procedure for solving fluid-structure-interaction problems, is based on a partitioning of the fluid-structure system into a fluidic part and a structural part. In fluid-structure interaction, on short time scales the fluid appears as an added mass to the structural operator, and the stability and convergence properties of the subiteration process depend significantly on the ratio of this apparent added mass to the actual structural mass. In the present paper, we establish that the added-mass effects corresponding to compressible and incompressible flows are fundamentally different. For a model problem, we show that on increasingly small time intervals, the added mass of a compressible flow is proportional to the length of the time interval, whereas the added mass of an incompressible flow approaches a constant. We then consider the implications of this difference in proportionality for the stability and convergence properties of the subiteration process, and for the stability and accuracy of loosely coupled staggered time-integration methods.* [DOI: 10.1115/1.3059565]

*Keywords: fluid-structure interaction, added-mass effect, compressible and incompressible flow, subiteration*

## 1 Introduction

The numerical simulation of the interaction of a flexible structure with a contiguous fluid flow is of critical importance to a multitude of applications, including the analysis of aero-elastic instabilities such as flutter in aerospace engineering [1,2] and the investigation of cardiovascular disorders, such as vulnerable plaques and aneurysms in biomechanics [3,4]. The basic iterative method for solving fluid-structure-interaction problems is subiteration. In the subiteration method, the fluid and solid subproblems are solved alternatingly, subject to complementary partitions of the interface conditions. In strongly coupled partitioned schemes, the subiteration process is repeated until convergence to a prescribed tolerance. Alternatively, the subiteration method can be used as a preconditioner, for instance to a Krylov-subspace method [5,6] or as a smoother in multigrid [7]. In loosely coupled (or staggered) time-integration schemes, the subiteration procedure is performed only once per time step [2,8,9].

On short time scales, the effect of the fluid on the structure can be represented as an added mass. The ratio of this apparent added mass to the structural mass is critical to the convergence and stability properties of the subiteration process. If the characteristic mass ratio exceeds 1, then the subiteration process is unstable; see, e.g., Ref. [10]. The added-mass effect of incompressible flows has recently been studied in Refs. [10–12]. Heuristic methods to account for the added-mass effect in fluid-structure-interaction computations with very light structures, such as large cargo parachutes, have been proposed in Refs. [13,14]. However, improved understanding of these effects in engineering computations would be beneficial. The added-mass effect of compressible flows is not well known. Moreover, despite the fact that there is a general concensus that the behavior of subiteration is distinctly different for compressible and incompressible flows, it appears that the precise distinction is not well understood. This incomplete understanding has been the source of many miscommunications with regard to the stability properties of subiteration, and with regard to the accuracy and stability of staggered time-integration schemes, which depend strongly on the stability characteristics of the underlying subiteration procedure.

In the present paper, we investigate the difference between the added-mass effects pertaining to compressible and incompressible flows, and we consider the implications for the stability and convergence of the subiteration process, and for the stability and accuracy of staggered time-integration methods. Based on a model problem, viz., a fluid flow on a semi-infinite domain over a flexible panel in 2D, we show that the added mass of a compressible flow is proportional to the length of the time step in the time-integration process, whereas the added mass of an incompressible flow approaches a constant as the time step vanishes. Consequently, regardless of the density of the fluid and the mass of the structure, the subiteration process is stable and convergent for compressible flows for sufficiently small time steps. For incompressible flows, this is not the case, and the subiteration method can remain unstable in the limit of vanishing time-step size. The distinct difference in the added-mass effect of compressible and incompressible flows and in the corresponding properties of the subiteration method, is caused by the fact that for compressible flows the displacement of the interface affects the fluid only in the immediate vicinity of the interface, on account of the finite speed of sound in compressible fluids, whereas for incompressible fluids the displacement of the interface induces a global perturbation in the fluid. This qualitative difference between compressible and incompressible fluids applies identically to other fluid-structure-interaction problems. It is therefore anticipated that the results of this paper generalize mutatis mutandis to other more complicated fluid-structure-interaction problems.

For incompressible flows, the model problem that we consider is a generalization of that in Ref. [10], in that we include convective and viscous effects. Our analysis conveys, however, that these effects are subordinate in the short time-scale limit and, hence, in

Contributed by the Applied Mechanics Division of ASME for publication in the JOURNAL OF APPLIED MECHANICS. Manuscript received December 4, 2007; final manuscript received May 22, 2008; published online January 15, 2009. Review conducted by Arif Masud.

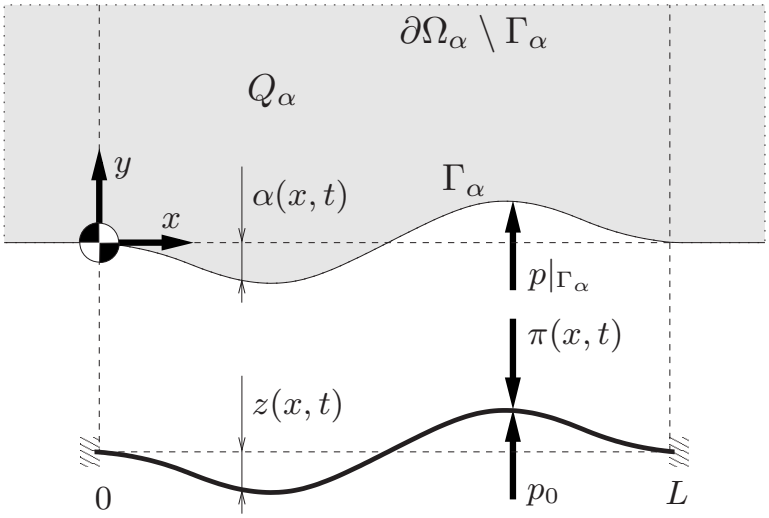



**Fig. 1 Illustration of the panel problem: temporal cross section with expanded interface region**

this limit we retrieve the results of Causin et al. [10] for incompressible flow. The approach in this paper is based on formal Fourier analyses of linearized model problems, without regard for convergence of the Fourier series in the appropriate norms. The results can be provided with a rigorous footing, but this is beyond the scope of the present paper.

The contents of this paper are organized as follows. Section 2 presents the problem statement. In Secs. 3 and 4 we derive the relation between the structural displacement and the corresponding pressure exerted by the fluid on the structure for the compressible-flow model and the incompressible-flow model, respectively. Section 5 investigates the stability and convergence properties of subiteration for the two flow types. In Sec. 6 we consider the implications of the distinct properties of subiteration for compressible and incompressible flows for the stability and accuracy of staggered time-integration methods. Section 7 contains concluding remarks.

## 2 Problem Statement

To formulate the model problems, let $x$, $y$, and $t$ designate a horizontal spatial coordinate, a vertical spatial coordinate, and a temporal coordinate, respectively. We consider an open space-time domain

$$Q_\alpha = \{(x,y,t): 0 < t < T,\ 0 < x < L,\ \alpha(x,t) < y < \infty\}$$

see the illustration in Fig. 1. The bottom boundary of $Q_\alpha$, which represents the *interface* between the compressible or incompressible fluid flow in $Q_\alpha$ and the structure, is given by

$$\Gamma_\alpha = \{(x,y,t): 0 < t < T,\ 0 < x < L,\ y = \alpha(x,t)\}$$

The fluid models are elaborated in Secs. 3 and 4.

The structural model that we consider pertains to the flexural vibration of a beam

$$m\frac{\partial^2 z}{\partial t^2} + \sigma^2\frac{\partial^4 z}{\partial x^4} = p_0 - \pi(x,t) \tag{1}$$

with $m$ as the mass of the beam per unit length, $z$ as the vertical displacement, $\sigma$ as the flexural rigidity, $p_0$ as the prescribed exterior pressure, and $\pi$ as the force exerted by the fluid on the structure.

Denoting by $p|_{\Gamma_\alpha}$ the pressure in the fluid at the interface, the fluid and the structure are connected by the dynamic and kinematic interface conditions

$$\pi(x,t) = p|_{\Gamma_\alpha}, \quad \alpha(x,t) = z(x,t) \tag{2}$$

The fluid-flow models associate a unique pressure field $\pi$ with each admissible interface displacement field $\alpha$. We refer to map $P: \alpha \mapsto \pi$ as the *displacement-to-pressure* (dtp) operator corresponding to a particular flow model. For the panel-model problem that we consider, the customary subiteration approach for solving fluid-structure-interaction problems can be condensed into the following iterative procedure. Given an initial approximation of the structural displacement, $z_0$, repeat for $n = 1, 2, \ldots$

$$m\frac{\partial^2 z_n}{\partial t^2} + \sigma^2\frac{\partial^4 z_n}{\partial x^4} = p_0 - P(z_{n-1}) \tag{3}$$

To elucidate the problem considered in this paper, let us consider the particular case that Eq. (1) is provided with the homogeneous initial conditions

$$z(x,0) = 0, \quad \partial_t z(x,0) = 0 \tag{4}$$

and, moreover, suppose that the flow problem is furnished with initial and boundary conditions such that it admits a uniform flow with pressure $p_0$. The obvious solution to Eq. (1) is then $z(x,t) = \bar{z}(x,t) = 0$, and the corresponding solution of the flow problem is the uniform flow specified by the initial conditions. By adding a suitable partition of zero to Eq. (3), we obtain

$$m\frac{\partial^2 (z_n - \bar{z})}{\partial t^2} + \sigma^2\frac{\partial^4 (z_n - \bar{z})}{\partial x^4} = -\left(P(z_{n-1}) - P(\bar{z})\right) \tag{5}$$

If we restrict our considerations to displacements that are small in the appropriate norm, the right member in Eq. (5) can be linearized, and we obtain the following recursion relation for the iteration error $\varepsilon_n = z_n - \bar{z}$ in the subiteration process:

$$m\frac{\partial^2 \varepsilon_n}{\partial t^2} + \sigma^2\frac{\partial^4 \varepsilon_n}{\partial x^4} = -P'\varepsilon_{n-1} \tag{6}$$

where $P'$ designates the linearized dtp operator. Moreover, under the stipulation that the iterates $z_n$ comply with the initial conditions, it follows that the iteration errors $\varepsilon_n$ satisfy homogeneous initial conditions

$$\varepsilon_n(x,0) = 0, \quad \partial_t \varepsilon_n(x,0) = 0 \tag{7}$$

In the sequel of this paper, we derive the linearized dtp operators for a compressible-flow and an incompressible-flow model, and we examine the corresponding behavior of the subiteration error in compliance with Eqs. (6) and (7).

## 3 Compressible Flow Model

We consider a compressible flow governed by the Euler equations

$$\frac{\partial \mathbf{q}}{\partial t}+\frac{\partial \mathbf{f}(\mathbf{q})}{\partial x}+\frac{\partial \mathbf{g}(\mathbf{q})}{\partial y}=0, \quad (x,y,t)\in Q_\alpha \tag{8a}$$

with

$$\mathbf{q}:=(q_1,q_2,q_3,q_4)$$

$$\mathbf{f}(\mathbf{q}):=\left(q_2,\frac{q_2^2}{q_1}+p(\mathbf{q}),\frac{q_2q_3}{q_1},\frac{q_2(p(\mathbf{q})+q_4)}{q_1}\right) \tag{8b}$$

$$\mathbf{g}(\mathbf{q}):=\left(q_3,\frac{q_3^2}{q_1}+p(\mathbf{q}),\frac{q_2q_3}{q_1},\frac{q_3(p(\mathbf{q})+q_4)}{q_1}\right)$$

In Eq. (8*b*), $q_1$, $q_2$, $q_3$, and $q_4$ denote the density, horizontal momentum, vertical momentum, and total energy of the fluid, respectively. The system (8) is closed by the equation of state

$$p(\mathbf{q}):=(\gamma-1)\left(q_4-\tfrac{1}{2}(q_2^2+q_3^2)/q_1\right) \tag{8c}$$

with $\gamma=1.4$.

At the interface, the fluid flow complies with the flow-tangency condition

$$\frac{\partial \alpha}{\partial t}+\frac{q_2}{q_1}\frac{\partial \alpha}{\partial x}-\frac{q_3}{q_1}=0 \tag{9}$$

The boundary conditions on the complement $\partial\Omega_\alpha\backslash\Gamma_\alpha$ will not be further elaborated.

To derive the linearized dtp operator corresponding to Eq. (8), we consider small deflections

$$\alpha_\epsilon=0+\epsilon\alpha', \quad \epsilon\to 0 \tag{10}$$

Accordingly, we assume that the fluid solution can be formally expanded as $\mathbf{q}_\epsilon=\mathbf{q}_0+\epsilon\mathbf{q}'+O(\epsilon^2)$, where the generating solution $\mathbf{q}_0$ corresponds to a uniform horizontal flow with density $\rho_0>0$, horizontal velocity $U_0\geq 0$, and pressure $p_0$

$$\mathbf{q}_0(x,y,t)=\left(\rho_0,\rho_0U_0,0,\tfrac{1}{2}\rho_0U_0^2+p_0/(\gamma-1)\right) \tag{11}$$

One easily verifies that Eq. (11) indeed satisfies Eqs. (8) and (9) for $\alpha=0$. In addition, we assume that $\mathbf{q}_\epsilon$ is isentropic and irrotational. The first-order perturbation in the fluid solution can then be written as $\mathbf{q}'=(\rho',\rho'U_0+\rho_0\partial_x\varphi',\rho_0\partial_y\varphi',E')$, where the potential $\varphi'$ complies with the linearized full-potential equation

$$U_0^2\frac{\partial^2\varphi'}{\partial x^2}+2U_0\frac{\partial^2\varphi'}{\partial x\,\partial t}+\frac{\partial^2\varphi'}{\partial t^2}-C_0^2\left(\frac{\partial^2\varphi'}{\partial x^2}+\frac{\partial^2\varphi'}{\partial y^2}\right)=0 \tag{12}$$

with $C_0:=\sqrt{\gamma p_0/\rho_0}$ as the speed of sound corresponding to the reference state. The energy perturbation $E'$ is irrelevant in the sequel. The density perturbation $\rho'$ it related to the potential by $\rho'=-(\rho_0/C_0^2)(\partial_t\varphi'+U_0\partial_x\varphi')$. Moreover, upon expanding the pressure according to $p(\mathbf{q}_\epsilon)=p(\mathbf{q}_0)+\epsilon p'+O(\epsilon^2)$, it holds that $p'=C_0^2\rho'$. The flow-tangency condition (9) yields the first-order condition

$$\frac{\partial \alpha'}{\partial t}+U_0\frac{\partial \alpha'}{\partial x}-\frac{\partial \varphi'}{\partial y}=0 \tag{13}$$

It is to be noted that Eqs. (12) and (13) hold in the unperturbed domain $Q_0$ and on the unperturbed interface $\Gamma_0$, respectively.

Green's function for the wave equation (see, for instance, Ref. [15], p. 473 and Ref. [16], p. 520) enables us to express the pressure perturbation at $\Gamma_0$ in accordance with Eqs. (12) and (13) as $p'|_{\Gamma_0}=P'\alpha'$, with operator $P'$ according to

$$P'=\pi^{-1}\rho_0C_0\Psi\Xi\Psi \tag{14a}$$

where

$$(\Xi\psi)(x,t)=\int_0^t\int_{\mathbb{R}}\psi(\xi,\tau)\times\frac{H(C_0(t-\tau)-|(x-\xi)-U_0(t-\tau)|)}{\sqrt{C_0^2(t-\tau)^2-|(x-\xi)-U_0(t-\tau)|^2}}d\xi d\tau \tag{14b}$$

with $H(\cdot)$ as the Heaviside function and

$$\Psi\varepsilon=\begin{cases}\partial_t\varepsilon+U_0\partial_x\varepsilon & \text{if } (x,t)\in]0,L[\times]0,T[\\ 0 & \text{otherwise}\end{cases} \tag{14c}$$

It is noteworthy that the Heaviside function restricts the domain of integration to the triangle

$$\{\tau<t, x-(U_0+C_0)(t-\tau)<\xi<x-(U_0-C_0)(t-\tau)\} \tag{15}$$

which constitutes the projection of the domain of dependence associated with Eq. (12) for the space/time coordinate $(x,t)$ onto $\Gamma_0$. Equation (14) represents the linearized dtp operator corresponding to the considered compressible-flow model.

To facilitate the interpretation of the added-mass effect associated to Eq. (14), we derive the Fourier symbol of the operator (14). To this end, we first derive the Fourier symbol of the integral operator $\Xi$. Let us consider an isolated Fourier mode

$$\psi(x,t)=\hat{\psi}(\kappa,\omega)\exp(i\kappa x+i\omega t) \tag{16}$$

Upon inserting Eq. (16) into Eq. (14*b*), and restricting the domain of integration in accordance with Eq. (15), we obtain $(\Xi\psi)(x,t)=\hat{\Xi}(\kappa,\omega,x,t)\hat{\psi}(\kappa,\omega)\exp(i\kappa x+i\omega t)$, where the Fourier symbol $\hat{\Xi}$ is given by

$$\hat{\Xi}(\kappa,\omega,x,t)=\int_0^t\int_{x-(U_0+C_0)(t-\tau)}^{x-(U_0-C_0)(t-\tau)}e^{-i(\kappa(x-\xi)+\omega(t-\tau))}\times(C_0^2(t-\tau)^2-|(x-\xi)-U_0(t-\tau)|^2)^{-1/2}d\xi d\tau \tag{17}$$

We introduce the transformations

$$(\theta,\eta)\mapsto(\xi,\tau)=(x-(U_0-C_0\sin\theta)\eta,t-\eta)$$
$$(r,\zeta)\mapsto(\kappa,\omega)=r2t^{-1}(C_0^{-1}\cos\zeta,\sin\zeta) \tag{18}$$

Note that the factor $C_0^{-1}$ is a prerequisite in the second transformation in Eq. (18) to ensure dimensional consistency. By means of Eq. (18) and the partition of unity $1=\sin^2\theta+\cos^2\theta$, the integral (17) can be condensed into

$$\hat{\Xi}=t\int_{-\pi/2}^{\pi/2}\frac{\sin\beta}{\beta}\exp(-i\beta)d\theta \tag{19}$$

where $\beta(r,\zeta,\theta)=r(\mathrm{M}\cos\zeta+\sin\zeta-\cos\zeta\sin\theta)$, with $\mathrm{M}=U_0/C_0$ as the Mach number. Noting that $|\beta^{-1}\sin\beta|\leq 1$ for all $\beta\in\mathbb{R}$, it follows from the Cauchy–Schwartz inequality that the Fourier symbol of $\Xi$ can be bounded as

$$|\hat{\Xi}(\kappa,\omega,x,t)|\leq t\|(\cdot)^{-1}\sin(\cdot)\|_{L^2(-\pi/2,\pi/2)}\times\|\exp(-i(\cdot))\|_{L^2(-\pi/2,\pi/2)}\leq\pi t \tag{20}$$

For the operator $\Psi$ according to Eq. (14*c*) we simply obtain $(\Psi\psi)(x,t)=\hat{\Psi}(\kappa,\omega)\hat{\psi}(\kappa,\omega)\exp(i\kappa x+i\omega t)$, with $\hat{\Psi}=i(\omega+U_0\kappa)$. The Fourier symbol of the composite operator (14*a*) is the product of the Fourier symbols of the operators in the composition. Hence, we obtain the following upper bound for the Fourier symbol of the linearized dtp operator (14) associated with the compressible-flow problem

$$|\hat{P}| \leq \rho_0 C_0 t|\omega + U_0\kappa|^2 \tag{21}$$

In particular, in the analysis of the added-mass effect, we shall be interested in short time intervals or, equivalently, high frequencies. In this context, it is to be noted that Eq. (21) yields $|\hat{P}| \lesssim \rho_0 C_0 t\omega^2$ in the high-frequency limit $\omega \to \infty$. The Fourier symbol of this high-frequency limit can be associated with an added mass

$$\mu_c = \rho_0 C_0 t \tag{22}$$

Hence, the added mass corresponding to the compressible flow is time dependent and, specifically, the added mass $\mu_c$ is proportional to $t$.

Let us allude to the fact that the added mass $\mu_c$ in Eq. (22) admits an intuitive physical interpretation: because pressure perturbations travel at the speed of sound $C_0$, the displacement of the interface has a *local* effect on the fluid, and only affects the fluid in a region within distance $C_0 t$ of the interface. The mass corresponding to this region (per unit length) is precisely $\mu_c$.

## 4 Incompressible Flow Model

We consider an incompressible flow governed by the Navier–Stokes equations

$$\partial_t u + \partial_x uu + \partial_y uv + \partial_x p - \nu\Delta u = 0 \tag{23a}$$

$$\partial_t v + \partial_x uv + \partial_y vv + \partial_y p - \nu\Delta v = 0 \tag{23b}$$

$$\partial_x u + \partial_y v = 0 \tag{23c}$$

where $u$ and $v$ represent the horizontal and vertical velocity components, respectively, $p$ denotes the pressure divided by the (homogeneous) fluid density $\rho_0$, $\nu$ is the dynamic viscosity, and $\Delta$ designates the Laplace operator.

At the interface, the flow is assumed to obey slip boundary conditions. This implies that the flow complies with the tangency condition

$$\frac{\partial\alpha}{\partial t} + u\frac{\partial\alpha}{\partial x} - v = 0 \tag{24}$$

and, moreover, that the tangential component of the normal traction vanishes

$$\mathbf{n}_\alpha \cdot \boldsymbol{\nabla}\mathbf{u} \cdot \mathbf{t}_\alpha + \mathbf{t}_\alpha \cdot \boldsymbol{\nabla}\mathbf{u} \cdot \mathbf{n}_\alpha = 0 \tag{25}$$

where $\mathbf{n}_\alpha$ and $\mathbf{t}_\alpha$ denote the unit normal vector and the unit tangential vector to $\Gamma_\alpha$, respectively, $\boldsymbol{\nabla} = (\partial_x, \partial_y)$, and $\mathbf{u} = (u, v)$. The boundary conditions on $\partial\Omega_\alpha \backslash \Gamma_\alpha$ will be elaborated in passing.

We are concerned with small deflections $\alpha_\epsilon$ conforming to Eq. (10) and, accordingly, we assume that the flow solution can be formally expanded as $(u,v,p)_\epsilon = (u,v,p)_0 + \epsilon(u,v,p)' + O(\epsilon^2)$, where the generating solution $(u,v,p)_0 = (U_0, 0, p_0)$ again corresponds to a uniform horizontal flow. Upon inserting the expansion in Eq. (23*a*) and collecting terms of $O(\epsilon)$, we obtain the first-order conditions

$$\partial_t u' + U_0\partial_x u' + \partial_x p' - \nu\Delta u' = 0 \tag{26a}$$

$$\partial_t v' + U_0\partial_x v' + \partial_y p' - \nu\Delta v' = 0 \tag{26b}$$

$$\partial_x u' + \partial_y v' = 0 \tag{26c}$$

These conditions hold on $Q_0$. The boundary conditions (24) and (25) moreover imply that $u'$ and $v'$ comply with the following first-order conditions on $\Gamma_0$:

$$\frac{\partial\alpha'}{\partial t} + U_0\frac{\partial\alpha'}{\partial x} - v' = 0, \quad \frac{\partial u'}{\partial y} + \frac{\partial v'}{\partial x} = 0 \tag{27}$$

For notational convenience, we introduce the condensed notation $\mathbf{q}'(x,y,t) = (u', v', p')(x,y,t)$. Instead of deriving an explicit expression for the linearized dtp operator corresponding to Eqs. (26) and (27), we establish its Fourier symbol. To this end, we regard an isolated Fourier component of the interface displacement

$$\alpha'(x,t) = \hat{\alpha}(\kappa,\omega)\exp(i\kappa x + i\omega t) \tag{28}$$

and a corresponding velocity/pressure perturbation

$$\mathbf{q}'(x,y,t) = \hat{\mathbf{q}}(k,\omega)\exp(i\kappa x + i\omega t + sy) \tag{29}$$

We stipulate that the velocity and pressure perturbations vanish as $y \to \infty$. This implies that the functions $s := s(\kappa,\omega)$ must have strictly negative real part. Upon inserting Eq. (29) into Eq. (26), we obtain $\hat{\mathbf{N}}(k,\omega)\cdot\hat{\mathbf{q}}(k,\omega)\exp(i\kappa x + i\omega t + sy) = 0$, where the Fourier symbol $\hat{\mathbf{N}}(k,\omega)$ of system (26) is defined by

$$\hat{\mathbf{N}}(k,\omega) = \begin{pmatrix} \hat{H}(k,\omega) & 0 & i\kappa \\ 0 & \hat{H}(k,\omega) & s \\ i\kappa & s & 0 \end{pmatrix} \tag{30}$$

with $\hat{H}(k,\omega) = i\omega + iU_0\kappa + \nu(\kappa^2 - s^2)$. Therefore, Eq. (29) complies with Eq. (26) if and only if $\hat{\mathbf{q}}(k,\omega) \in \text{kernel}(\hat{\mathbf{N}}(k,\omega))$. This equation admits nontrivial solutions under the strict condition that $\det(\hat{\mathbf{N}}(k,\omega)) = (\kappa^2 - s^2)\hat{H}(k,\omega) = 0$. It then follows that Eq. (29) satisfies Eq. (26) provided that

$$\begin{gathered} \hat{\mathbf{q}}(k,\omega) \in \text{span}\{(i\kappa, -|\kappa|, -i(\omega + U_0\kappa))\}, \quad s = -|\kappa|, \quad \text{or} \\ \hat{\mathbf{q}}(k,\omega) \in \text{span}\{(s, -i\kappa, 0)\}, \quad \hat{H}(k,\omega) = 0 \end{gathered} \tag{31}$$

A solution to Eqs. (26) and (27) with $\alpha'$ specified by Eq. (28) can be obtained by combining the modes (Eq. (31))

$$\begin{aligned} \mathbf{q}' = \hat{\alpha}\Bigg[ & \frac{(\omega + U_0 k)(\sigma^2 + \kappa^2)}{|\kappa|(\kappa^2 - \sigma^2)}\exp(-|\kappa|y)\begin{pmatrix} -\kappa \\ -i|\kappa| \\ (\omega + U_0\kappa) \end{pmatrix} \\ & + \frac{2(\omega + U_0 k)\kappa}{\kappa^2 - \sigma^2}\exp(\sigma y)\begin{pmatrix} -\sigma \\ i\kappa \\ 0 \end{pmatrix}\Bigg]\exp(i\kappa x + i\omega t) \end{aligned} \tag{32}$$

with $\sigma(\kappa,\omega) = \pm\sqrt{\kappa^2 + i(\omega + U_0\kappa)/\nu}$, subject to the restriction that the real part of $\sigma$ is negative. Recalling that the pressure divided by the density corresponds to the third component in Eq. (32), we obtain the following Fourier symbol for the linearized dtp operator corresponding to the incompressible flow

$$\hat{P}(\kappa,\omega) = \rho_0\left(-\frac{(\omega + U_0 k)^2}{|\kappa|} + i\frac{2\nu\kappa^2(\omega + U_0\kappa)}{|\kappa|}\right) \tag{33}$$

It is to be noted that the high-frequency limit of Eq. (33) yields $\hat{P} \sim -\rho_0|\kappa|^{-1}\omega^2$ as $\omega \to \infty$. This symbol can be associated with an added mass $\rho_0|\kappa|^{-1}$. In fact, the wave number can only assume values $\kappa = k\pi/L$, $k \in \mathbb{N}$ on account of the structural boundary conditions $\alpha(0,t) = \alpha(L,t) = 0$. Hence, the largest-wavelength component ($k = 1$) is dominant, and for this component it holds that $\hat{P} \sim -\mu_i\omega^2$ as $\omega \to \infty$, where the added mass is defined by

$$\mu_i = \rho_0 L/\pi \tag{34}$$

Equation (34) conveys that the added mass corresponding to the incompressible flow is independent of time. It is noteworthy that added mass (34) is consistent with that derived in Ref. [10], in the appropriate limit.

To provide a physical explanation for the difference in the added-mass effect for compressible and incompressible flows, we note that mode (32) is *global*. Hence, whereas for compressible flows the effect of the displacement of the interface on the fluid is confined to a region within distance $C_0 t$ of the interface (see Sec.

3), for incompressible flows the fluid is affected throughout its entire domain.

Let us moreover note that the convective part and the viscous part of $\hat{P}$ according to Eq. (33) are proportional to $\omega$, whereas the added-mass part is proportional to $\omega^2$. Hence, convective effects and viscous effects are subordinate to the added-mass effect in the limit $\omega \to \infty$.

## 5 Stability and Convergence of Subiteration

Equipped with the Fourier symbols of the linearized dtp operators, we can establish the behavior of the iteration error according to Eq. (6) for the compressible and incompressible flows. Let us consider an isolated Fourier component of the iteration error: $\varepsilon_n(x,t)=\hat{\varepsilon}_n(\kappa,\omega)\exp(i\kappa x+i\omega t)$. Upon inserting this component into Eq. (6), we obtain the relation $|\hat{\varepsilon}_n(\kappa,\omega)| \leq \varrho(\kappa,\omega)|\hat{\varepsilon}_{n-1}(\kappa,\omega)|$, where the contraction number $\varrho$ is defined by

$$\varrho(\kappa,\omega)=\frac{|\hat{P}(\kappa,\omega)|}{|-m\omega^2+\sigma^2\kappa^4|} \tag{35}$$

Again restricting our consideration to high frequencies, it follows that the contraction number is bounded from above as $\varrho \leq \mu/m$ as $\omega \to \infty$, where $\mu$ refers to the added mass according to Eqs. (22) and (34) for the compressible flow and incompressible flow, respectively, and equality holds in the incompressible case. Let us note that the following results extend without further modifications to other structural-stiffness operators, as for fixed $\kappa$ the contribution corresponding to the structural-stiffness operator to Eq. (35) vanishes in the limit $\omega \to \infty$. This argument has also been used in Ref. [12]. If $\varrho \leq 1$, the Fourier amplitudes $\hat{\varepsilon}_n$ form a nonincreasing sequence and, hence, the subiteration process is stable. Moreover, if $\varrho < 1$, the subiteration process is formally convergent, and $\varrho$ determines the rate of convergence. For the compressible and incompressible flows, Eqs. (22), (34), and (35) lead to the following estimates for the corresponding contraction numbers:

$$\varrho_c \leq \frac{\rho_0 C_0 t}{m}+O(\omega^{-1}), \quad \varrho_i=\frac{\pi^{-1}\rho_0 L}{m}+O(\omega^{-1}) \tag{36}$$

as $\omega \to \infty$.

The estimates in Eq. (36) elucidate the fundamental difference in the properties of the subiteration method for compressible and incompressible flows. In computational methods, the subiteration procedure is generally applied to resolve the aggregated fluid-structure system within each time step of a time-integration process, i.e., iteration (3) is repeated within each time step until the iteration error is inferior to a certain prescribed tolerance. Hence, within a time step, the sequence of iteration errors complies with Eqs. (6) and (7), and we implicitly restrict our consideration of the iteration error to the time interval $0 \leq t \leq \delta t$, where $\delta t$ denotes the time step in the time-integration process. The upper bound $\varrho_c$ in Eq. (36) then yields $\varrho_c \leq \rho_0 C_0 \delta t/m$. In particular, this implies that for compressible flows the convergence behavior of the subiteration process improves if the time step is reduced and, specifically, $\varrho_c \to 0$ as $\delta t \to 0$. Let us remark that this behavior has also been established for the piston problem in Ref. [17]. Consequently, for all settings of the structural mass $m$ and the fluid density $\rho_0$, there exists a strictly positive time step $\delta t^*$ such that the subiteration process is stable for all $\delta t \in ]0, \delta t^*]$. Moreover, if the time-step size is reduced by a certain factor, then the convergence rate of the subiteration process improves by that same factor. For incompressible flows, this is not the case. For increasingly small time steps, i.e., in the limit $\delta t \to 0$, the contraction number converges toward the strictly positive, time-step-independent high-frequency limit in Eq. (36). Therefore, if the characteristic fluid-structure mass ratio $\mu_i/m$ exceeds 1, the subiteration method is unstable, regardless of the time step.[1]

The above results have been established on the basis of the continuum problem. If a particular temporal discretization scheme is considered, then the structure of the estimates in Eq. (36) remains intact, although the precise values can be different. We refer to Ref. [12] for an overview of the effects of temporal discretization schemes on the stability of the subiteration procedure for fluid-structure interaction with incompressible flow.

## 6 Staggered Time-Integration Methods

The aforementioned fundamental difference in the convergence properties of the subiteration process for compressible and incompressible flows also carries important consequences for the suitability of staggered (also referred to as loosely coupled or partitioned) time-integration procedures, i.e., time-integration methods in which the subiteration step is performed only once per time step; see, for instance, Refs. [2,8,9]. We regard a partition of the time interval under consideration, $0<t<T$, into time steps $t_{i-1}<t<t_i$ of uniform length $\delta t=t_i-t_{i-1}$ $(i=1,2,\ldots,T/\delta t)$. Within each time step, the aggregated fluid-structure system can be condensed into

$$\mathcal{A}w_i=\mathcal{B}w_{i-1} \quad \text{with} \quad w_i=\begin{pmatrix} q_i \\ z_i \end{pmatrix}, \quad \mathcal{A}=\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \quad \mathcal{B}=\begin{pmatrix} B_{11} & 0 \\ 0 & B_{22} \end{pmatrix} \tag{37}$$

where $q_i$ and $z_i$ represent the variables pertaining to the discrete approximation of the fluid and structure solutions on interval $i$, and $A_{11}$, $A_{12}$, $A_{21}$, and $A_{22}$ denote the discretized fluid operator, kinematic condition, dynamic condition, and structural operator, respectively. The operators $B_{11}$ and $B_{22}$ extract the initial conditions for the fluid and structure subsystems on interval $i$ from the approximation on the previous time interval. Of course, on the first interval the right member in Eq. (37) is replaced with a vector corresponding to the prescribed initial conditions. For simplicity, we assume that the operators $\mathcal{A}$ and $\mathcal{B}$ are linear, which is appropriate for the ensuing error analysis.

Let us assume that system (37) has been solved inexactly on the previous time interval, $i-1$. In particular, the result on interval $i-1$ contains an error $\delta w_{i-1}$. This error propagates to an error $\delta w_{p,i}$ on interval $i$ via the initial conditions. Hence, on account of the inexact solution on interval $i-1$, Eq. (37) is replaced with

$$\mathcal{A}(w_i+\delta w_{p,i})=\mathcal{B}(w_{i-1}+\delta w_{i-1}) \tag{38}$$

By virtue of the assumed linearity of Eq. (38), the propagated error can be expressed in terms of the error on interval $i-1$ as $\delta w_{p,i}=\mathcal{L}\delta w_{i-1}$ with $\mathcal{L}=\mathcal{A}^{-1}\mathcal{B}$. Note that the inverse operator $\mathcal{A}^{-1}$ is well defined under the standing assumption that the fluid-structure problem is well posed.

Application of the subiteration procedure to Eq. (38) leads to the following sequence of approximations. Given an initial estimate $w_{i,0}$, for $n=1,2,\ldots$

$$\begin{pmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{pmatrix}\begin{pmatrix} q_{i,n} \\ z_{i,n} \end{pmatrix}=\begin{pmatrix} B_{11} & 0 \\ 0 & B_{22} \end{pmatrix}\begin{pmatrix} q_{i-1}+\delta q_{i-1} \\ z_{i-1}+\delta z_{i-1} \end{pmatrix}-\begin{pmatrix} 0 & A_{12} \\ 0 & 0 \end{pmatrix}\begin{pmatrix} q_{i,n-1} \\ z_{i,n-1} \end{pmatrix} \tag{39}$$

Note that the fluid and structure approximations with index $n$, in fact, depend exclusively on the structural approximation with in-

[1] In principle, this statement requires somewhat more care because it is not a priori obvious that $\varrho_i$ in Eq. (36) does not represent an upper bound attained in the limit $\omega \to \infty$. A more precise analysis of Eq. (35) with $\hat{P}$ according to Eq. (33) reveals that this is not the case.

dex $n-1$. Hence, to initialize the procedure, it is sufficient to prescribe $z_{i,0}$. We define the local iteration error by $\delta w_{i,n}=w_{i,n}-(w_i+\delta w_{p,i})$. Upon adding a suitable partition of zero to Eq. (39), we obtain the error-amplification relation

$$\begin{pmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{pmatrix}\begin{pmatrix} \delta q_{i,n} \\ \delta z_{i,n} \end{pmatrix} = -\begin{pmatrix} 0 & A_{12} \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \delta q_{i,n-1} \\ \delta z_{i,n-1} \end{pmatrix} \tag{40}$$

From Eq. (40), it follows that $\delta w_{i,n}=\mathcal{Q}\delta w_{i,n-1}$ with

$$\mathcal{Q}=\begin{pmatrix} 0 & -A_{11}^{-1}A_{12} \\ 0 & A_{22}^{-1}A_{21}A_{11}^{-1}A_{12} \end{pmatrix} \tag{41}$$

Hence, by recursion, $\delta w_{i,n}=\mathcal{Q}^n\delta w_{i,0}$.

Suppose that the initial approximation on each time interval is obtained by means of prediction, i.e., by extrapolation of the approximation on the previous time interval. In particular

$$w_{i,0}=\mathcal{E}(w_{i-1}+\delta w_{i-1}) \tag{42}$$

where $\mathcal{E}$ represents the extension operator: $(\mathcal{E}w_{i-1})(x,t)=w_{i-1}(x,t+\delta t)$ for $0<t<\delta t$. The extension is well defined for finite-element approximations in time. For finite-difference approximations, it can be defined via interpolating polynomials. Assuming that in each time step the subiteration process is terminated after $\bar{n}$ iterations, the cumulative iteration error $\delta w_i$ in the final result on interval $i$ is composed of the propagated error and the local iteration error at iteration $\bar{n}$. From Eqs. (38)–(42), we then obtain the sequence of identities

$$\begin{aligned}\delta w_i &= \delta w_{i,\bar{n}}+\delta w_{p,i}=\mathcal{Q}^{\bar{n}}\delta w_{i,0}+\delta w_{p,i} \\ &= \mathcal{Q}^{\bar{n}}(\mathcal{E}(w_{i-1}+\delta w_{i-1})-(w_i+\delta w_{p,i}))+\delta w_{p,i} \\ &= \mathcal{Q}^{\bar{n}}(\mathcal{E}-\mathcal{L})w_{i-1}+(\mathcal{Q}^{\bar{n}}(\mathcal{E}-\mathcal{L})+\mathcal{L})\delta w_{i-1}\end{aligned} \tag{43}$$

The final identity in Eq. (43) is a consequence of $w_i=\mathcal{L}w_{i-1}$ and $\delta w_{p,i}=\mathcal{L}\delta w_{i-1}$.

From Eq. (43) it follows by recursion that

$$\delta w_i=\sum_{k=1}^{i}(\mathcal{Q}^{\bar{n}}(\mathcal{E}-\mathcal{L})+\mathcal{L})^{i-k}\mathcal{Q}^{\bar{n}}(\mathcal{E}-\mathcal{L})w_{k-1} \tag{44}$$

and, by the triangle inequality

$$\|\delta w_i\|=\sum_{k=1}^{i}\|\mathcal{Q}^{\bar{n}}(\mathcal{E}-\mathcal{L})+\mathcal{L}\|^{i-k}\|\mathcal{Q}\|^{\bar{n}}\|\mathcal{E}-\mathcal{L}\|\|w_{k-1}\| \tag{45}$$

Recalling that $t_i=i\delta t$, we replace $i-k$ in the exponent in Eq. (45) with $(t_i-t_k)/\delta t$. A necessary condition for boundedness of the right member in Eq. (45) in the limit $\delta t\to 0$ is

$$\|\mathcal{Q}^{\bar{n}}(\mathcal{E}-\mathcal{L})+\mathcal{L}\|\le 1+\vartheta\delta t \quad \text{as} \quad \delta t\to 0 \tag{46}$$

for some positive constant $\vartheta$. The exponential term in Eq. (45) can then be bounded as

$$\|\mathcal{Q}^{\bar{n}}(\mathcal{E}-\mathcal{L})+\mathcal{L}\|^{i-k}\le(1+\vartheta\delta t)^{(t_i-t_k)/\delta t}\sim e^{\vartheta(t_i-t_k)} \tag{47}$$

as $\delta t\to 0$. It is to be remarked that provision (46) does not hold for $\bar{n}=0$ because any appropriate norm of the extrapolation operator $\|\mathcal{E}\|$ exceeds 1 as $\delta t\to 0$. In particular, this implies that the analysis below does not hold if only extension is applied, or if the subiteration process is nonconvergent or if convergence is too slow. In such circumstances, the right member in Eq. (45) becomes unbounded as $\delta t\to 0$.

Proceeding under assumption (46), it follows from Eqs. (45) and (47) that

$$\|\delta w_i\|\le C\delta t^{-1}e^{\vartheta t_i}\|\mathcal{Q}\|^{\bar{n}}\|\mathcal{E}-\mathcal{L}\|\sup_k\|w_k\| \tag{48}$$

for some constant $C$ independent of $\delta t$, as $\delta t\to 0$. Suppose that the extension operator corresponds to an $m$th order extrapolation. Then for sufficiently smooth functions $\|\mathcal{E}-\mathcal{L}\|=O(\delta t^m)$. Moreover, on account of the fact that $\|w_k\|$ pertains to a time interval of length $\delta t$, it holds that $\|w_k\|=O(\delta t^{1/2})$. Therefore

$$\|\delta w_i\|\le\bar{C}(t_i)\|\mathcal{Q}\|^{\bar{n}}O(\delta t^{m-1/2}) \tag{49}$$

for some exponentially increasing function $\bar{C}(t)$, independent of $\delta t$.

The error $\delta w_i$, which is induced by the inexact solution of the aggregated fluid-structure system on the intervals with index $\le i$, is to be compared with the discretization error on interval, i.e., the difference between the resolved (monolithic) discrete solution, and the actual continuum solution. Suppose that the monolithic discrete approximation corresponding to Eq. (37) yields an approximation to the solution of the fluid-structure system with formal temporal order of accuracy $m$, i.e., for sufficiently smooth solutions its holds that the approximation error on each time interval conforms to

$$\|w_i-\bar{w}\|\le C\delta t^m\|\bar{w}\|=O(\delta t^{m+1/2}) \tag{50}$$

as $\delta t\to 0$, where $\bar{w}$ represents the continuum solution. The additional factor 1/2 in estimate (50) originates from the fact that the measure of the considered time interval is proportional to $\delta t$.

The upper bound (49) enables us to clarify the distinctly different properties of staggered time-integration procedures for compressible and incompressible flows. For compressible flows, $\|\mathcal{Q}\|$ is proportional to $\delta t$. In Sec. 5 this proportionality has been established for the map $\delta z_{i,n-1}\mapsto\delta z_{i,n}$, cf. Eq. (36). However, specifically, the norm of the map between the structure displacement and the fluid state, $\delta z_{i,n-1}\mapsto\delta q_{i,n}$, is proportional to $\delta t$, and the norm of the map between the fluid state and the structure displacement, $\delta q_{i,n}\mapsto\delta z_{i,n}$, is proportional to 1 as $\delta t\to 0$. Upon inserting the proportionality $\|\mathcal{Q}\|\propto\delta t$ into Eq. (49), it follows that for a compressible flow the iteration error on interval $i$, i.e., the error relative to the monolithic result, is bounded as: $\|\delta w_i\|\le C_a(t_i)\delta t^{m+\bar{n}-1/2}$ as $\delta t\to 0$, for some exponentially increasing function $C_a(t)$, independent of $\delta t$. For a staggered time-integration method, $\bar{n}=1$ and, therefore, the cumulative iteration error is of the same order as the discretization error in the monolithic result, cf. Eq. (50). Hence, the staggered procedure possesses the same order of accuracy as the underlying monolithic method, but with a different constant of proportionality. As a digression, we note that for $\bar{n}=2$, the cumulative iteration error is one order higher than the discretization error. Consequently, in the limit $\delta t\to 0$, the result obtained with two subiterations per time step is identical to the monolithic results.

For incompressible flows, staggered time-integration methods behave distinctly different. In the incompressible case, the norm of $\|\mathcal{Q}\|$ converges to a positive constant in the limit $\delta t\to 0$. For $\bar{n}=1$, the global iteration error thus remains $O(\delta t^{m-1/2})$ and, hence, the order of accuracy of a result obtained by a staggered method is one order lower than that of the underlying monolithic method. In fact, assuming that the subiteration process is convergent, the number of subiterations per time step must increase as $\bar{n}\propto|\log\delta t|$ as $\delta t\to 0$ to obtain a method, which yields the same order of accuracy as a monolithic approach.

The distinct properties of $\|\mathcal{Q}\|$ for compressible and incompressible flows is also pertinent in relation to condition (46). For compressible flows, $\|\mathcal{Q}\|\propto\delta t$ in the limit $\delta t\to 0$. Therefore, condition (46) is fulfilled for $\bar{n}\ge 1$ under the solitary provision that $\|\mathcal{L}\|=1+O(\delta t)$ as $\delta t\to 0$, independent of the extrapolation operator. This implies that if this provision holds, then the solution of the staggered scheme cannot grow unbounded in finite time, on account of upper bound (47). For incompressible flows, this is not the case because $\|\mathcal{Q}\|$ does not vanish as $\delta t\to 0$.

## 7 Conclusion

To examine the difference between the added-mass effects of compressible and incompressible flows, we considered the model

problem of flow in a semi-infinite domain over a flexible panel in 2D. We derived the displacement-to-pressure operator, which relates the pressure exerted by the fluid on the structure to the structural displacement for a compressible flow governed by the Euler equations and for an incompressible flow governed by the Navier–Stokes equations. For the compressible flow, the displacement-to-pressure operator assumes the form of an integrodifferential operator. We derived the Fourier symbol of this operator, and we showed that in the high-frequency limit corresponding to short time intervals, this Fourier symbol can be associated with an added mass proportional to the length of the considered time interval. For the incompressible flow, the Fourier symbol represents a time-independent added mass in the high-frequency limit. Moreover, we showed for the incompressible flow that the viscous and convective effects are subordinate to the added-mass effect in the high-frequency limit.

The distinct proportionalities of the added mass to the time step for compressible and incompressible flows yield essentially different behavior of the subiteration method for fluid-structure-interaction problems. For compressible flows, for any setting of the density of the fluid and the mass of the structure, the subiteration process is stable and convergent for sufficiently small time steps. Furthermore, if the time step in the time-integration method is reduced by a certain factor, then the convergence rate of the subiteration method improves by that same factor. For incompressible flows this is not the case, and the subiteration method can be unstable even in the limit of vanishing time-step size.

Finally, we considered the implications of the difference in the convergence behavior of the subiteration method for staggered time-integration methods. We showed that for compressible flows, the order of accuracy of a staggered method is identical to that of the underlying monolithic method, provided that a suitable predictor is used. If two subiterations per time step are applied instead of one, then the approximation provided by the staggered method approaches the monolithic result in the limit of vanishing time-step size. Moreover, we showed that for compressible flows, staggered time-integration methods are stable in the limit of vanishing time-step size, in the sense that the solution remains bounded in finite time. For incompressible flows, the order of accuracy of a stable staggered approximation with prediction is one order lower than the corresponding monolithic result. Moreover, for incompressible flows, time-integration schemes with a finite number of subiterations per time step can be unstable in the limit of vanishing time-step size, in the sense that the approximation can grow unbounded in finite time, if the subiteration process converges too slowly. Staggered methods therefore appear appropriate for fluid-structure-interaction problems with compressible flows, but for fluid-structure-interaction problems with incompressible flows their use should be dissuaded.

## References


[1] Farhat, C., Geuzaine, P., and Brown, G., 2003, "Application of a Three-Field Nonlinear Fluidstructure Formulation to the Prediction of the Aeroelastic Parameters of an f-16 Fighter," Comput. Fluids, **32**, pp. 3–29.

[2] Farhat, C., 2004, "CFD-Based Nonlinear Computational Aeroelasticity," *Encyclopedia of Computational Mechanics*, Vol. 3: Fluids, E. Stein, R. Borst, and T. Hughes, eds., Wiley, New York, pp. 459–480.

[3] Torii, R., Oshima, M., Kobayashi, T., Takagi, K., and Tezduyar, T., 2006, "Computer Modeling of Cardiovascular Fluid-Structure Interaction With the Deforming-Spatial-Domain/Stabilized-Space-Time Formulation," Comput. Methods Appl. Mech. Eng., **195**, pp. 1885–1895.

[4] Tezduyar, T., Sathe, S., Cragin, T., Nanna, B., Conklin, B., Pausewag, J., and Schwaab, M., 2007, "Modeling of Fluid-Structure Interactions With the Space-Time Finite Elements: Arterial Fluid Mechanics," Int. J. Numer. Methods Fluids, **54**, pp. 901–922.

[5] Michler, C., van Brummelen, H., and de Borst, R., 2006, "Error-Amplification Analysis of Subiteration-Preconditioned GMRES for Fluid-Structure Interaction," Comput. Methods Appl. Mech. Eng., **195**, pp. 2124–2148.

[6] Heil, M., 2004, "An Efficient Solver for the Fully-Coupled Solution of Large-Displacement Fluid-Structure Interaction Problems," Comput. Methods Appl. Mech. Eng., **193**, pp. 1–23.

[7] van Brummelen, H., van der Zee, K., and de Borst, R., 2008, "Space/Time Multigrid for a Fluid-Structure-Interaction Problem," Appl. Numer. Math., **58**(12), pp. 1951–1971.

[8] Piperno, S., and Farhat, C., 2001, "Partitioned Procedures for the Transient Solution of Coupled Aeroelastic Problems—Part II: Energy Transfer Analysis and Three-Dimensional Applications," Comput. Methods Appl. Mech. Eng., **190**, pp. 3147–3170.

[9] Felippa, C., Park, K., and Farhat, C., 2001, "Partitioned Analysis of Coupled Mechanical Systems," Comput. Methods Appl. Mech. Eng., **190**, pp. 3247–3270.

[10] Causin, P., Gerbeau, J., and Nobile, F., 2005, "Added-Mass Effect in the Design of Partitioned Algorithms for Fluid-Structure Problems," Comput. Methods Appl. Mech. Eng., **194**, pp. 4506–4527.

[11] LeTallec, P., and Mouro, J., 2001, "Fluid Structure Interaction With Large Structural Displacements," Comput. Methods Appl. Mech. Eng., **190**, pp. 3039–3067.

[12] Förster, C., Wall, W., and Ramm, E., 2007, "Artificial Added Mass Instabilities in Sequential Staggered Coupling of Nonlinear Structures and Incompressible Viscous Flows," Comput. Methods Appl. Mech. Eng., **196**, pp. 1278–1293.

[13] Tezduyar, T., Sathe, S., Keedy, R., and Stein, K., 2006, "Space-Time Finite Element Techniques for Computation of Fluid-Structure Interactions," Comput. Methods Appl. Mech. Eng., **195**, pp. 2002–2027.

[14] Tezduyar, T., 2006, "Interface-Tracking and Interface-Capturing Techniques for Finite Element Computation of Moving Boundaries and Interfaces," Comput. Methods Appl. Mech. Eng., **195**, pp. 2983–3000.

[15] Zauderer, E., 1989, *Partial Differential Equations of Applied Mathematics*, (Pure and Applied Mathematics), 2nd ed., Wiley, Chichester, West Sussex, UK.

[16] Haberman, R., 1998, *Applied Partial Differential Equations*, 3rd ed., Pearson Prentice-Hall, Upper Saddle River, NJ.

[17] van Brummelen, H., and de Borst, R., 2005, "On the Nonnormality of Subiteration for a Fluid-Structure Interaction Problem," SIAM J. Sci. Comput. (USA), **27**, pp. 599–621.

**Shu Takagi**[1]
Research Program for Computational Science,
RIKEN,
2-1 Hirosawa, Wako,
Saitama 351-0198, Japan
e-mail: takagish@riken.jp

**Takeshi Yamada**[2]
Department of Mechanical Engineering,
The University of Tokyo,
7-3-1 Hongo, Bunkyo-ku,
Tokyo 113-8656, Japan
e-mail: yuushi@fel.t.u-tokyo.ac.jp

**Xiaobo Gong**
Research Program for Computational Science,
RIKEN,
2-1 Hirosawa, Wako,
Saitama 351-0198, Japan
e-mail: gong@riken.jp

**Yoichiro Matsumoto**
Department of Mechanical Engineering,
The University of Tokyo,
7-3-1 Hongo, Bunkyo-ku,
Tokyo 113-8656, Japan
e-mail: ymats@mech.t.u-tokyo.ac.jp

# The Deformation of a Vesicle in a Linear Shear Flow

*In this paper, we discuss the motion of a vesicle in a linear shear flow. It is known that deformable vesicles such as liposomes show the so-called tank-treading and tumbling motions depending on the viscosity ratio between the inside and outside of the vesicle, the swelling ratio, and so on. First, we have conducted numerical simulations on the tank-treading motion of a liposome in a linear shear flow and compared the results with other numerical and experimental results. It is confirmed that the inclination angle of the vesicle becomes smaller when the viscosity ratio becomes larger or the swelling ratio becomes smaller and that the present results show quantitatively good agreement with other results. Then, the effects of membrane modeling are discussed from the mechanics point of view. There are two types of modeling for the lipid bilayer biomembrane. One is a two-dimensional fluid membrane, which reflects the fluidity of the lipid molecules. The other is a hyperelastic membrane, which reflects the stiffness of cytoskeleton structure. Liposome is usually modeled as a fluid membrane and red blood cell (RBC) is modeled as a hyperelastic one. We discuss how these differences of membrane models affect the behaviors of vesicles under the presence of shear flow. It is shown that the hyperelastic membrane model for RBC shows a less inclination angle of tank-treading motion and early transition from tank-treading to tumbling.* [DOI: 10.1115/1.3062966]

*Keywords: immersed boundary method, liposome, red blood cell, tank-treading, tumbling, membrane*

## 1 Introduction

In the capillary vessels, vesicles such as red blood cells (RBCs), drag delivery agents, and contrast agents change their shapes in response to the local flow conditions. In a macroscopic point of view, the deformation of these vesicles is related to the capability of passage through capillaries, and in the microscopic point of view, it is related to the efficiency of molecular transfer between the inside and outside of the vesicles. Therefore, to predict the mass transfer such as oxygen transfer from RBC to the surrounding tissues through capillaries, it is essential to understand the multiscale nature of the phenomena from the molecular scale to the continuum one. From these facts, the multiscale analysis for a microcirculation system becomes important.

In the case of biomembrane system in nature, molecular transfer across the membrane is very complicated through ion channel, protein driven pores, etc., and it is difficult to conduct the so-called multiscale simulation on these types of phenomena. Hence, instead of investigating such a complicated membrane, an artificial biomembrane called liposome, which has a simple bilayer structure of lipid molecules, can be used to develop the simulation methodology for the multiscale analysis in biomembrane system. Not only the methodological point of view but also the practical point of view, the multiscale simulation for liposome is useful to design its functions because liposome is expected to be utilized for the drag delivery agent, artificial oxygen carrier, etc. Since the experimental methods to produce liposomes and to investigate their characteristics are reasonably well developed, there is also an advantage that validation of the simulation results can be conducted through the comparison with liposomes whose characteristics are well controlled.

Because of the above-mentioned facts, we have been working on the multiscale analysis of liposome and numerically investigated molecular transfers across a lipid bilayer membrane by molecular dynamics simulations [1], mesocospic membrane properties by coarse grained type simulation [2], and macroscopic membrane deformation by continuum mechanics simulations, which are discussed in this paper.

Related to the membrane deformation of a liposome, there is an interesting phenomenon observed in a shear flow. A liposome in a linear shear flow shows two different types of the motions: tank-treading and tumbling. These motions are illustrated in Fig. 1. Tank-treading motion shows the caterpillarlike motion: steady shape with inclination angle $\theta$ constant having the surface velocity. Tank-treading motion occurs when the viscosity inside the vesicle is sufficiently small compared with that outside the vesicle. And, tumbling motion occurs when the viscosity inside the vesicle is sufficiently large. Increasing the internal viscosity makes the vesicles inclined in lower angle and it ends up with the tumbling motion below inclination angle $\theta=0$. Beaucourt et al. [3] conducted two-dimensional numerical simulations on the tank-treading motion of a liposome in a linear shear flow. They discuss the relation between the energy dissipation in the system and the motion change from tank-treading to tumbling. Kraus et al. [4] conducted three-dimensional simulation and obtained the tank-treading motion using an area preserving fluid membrane model.

A large number of numerical studies have been done related to the deformation of vesicles. Especially, simulations for RBC using particle methods are getting more popular [5,6]. Since we are interested in the multiscale approach using partial differential equations for the continuum mechanics, here we do not review these particle methods. Instead, we review some of the similar method to our approach, using Eulerian description with partial differential equations. A deformation of a vesicles in a shear flow

[1]1Corresponding author. Also at Department of Mechanical Engineering, The University of Tokyo.

[2]2Present address: Hitachi Company, 7-2-1 Omika-cho, Hitachi 319-1221, Japan.

Contributed by the Applied Mechanics Division of ASME for publication in the JOURNAL OF APPLIED MECHANICS. Manuscript received February 29, 2008; final manuscript received July 7, 2008; published online January 15, 2009. Review conducted by Arif Masud.

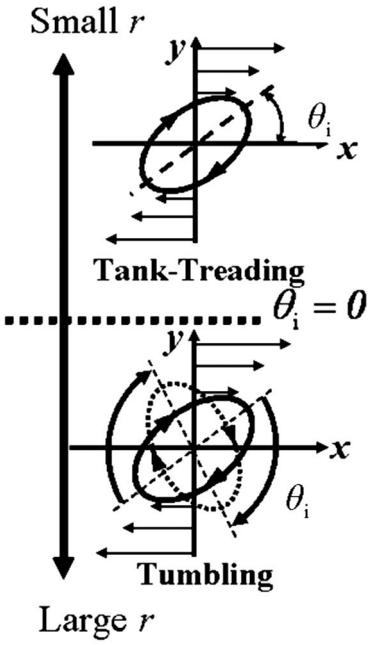



**Fig. 1 The behavior of deformable liposome in a simple shear flow (*r* is the viscosity ratio between the inside and outside of a liposome, and $\theta_i$ is the inclination angle)**

was investigated for the analysis of a RBC by several researchers [7–10]. They expressed a RBC membrane as a hyperelastic material. And various models such as neo-Hookean model and Evans–Skalak (ES) model were proposed to express the complex RBC membrane and have been investigated. In these models, the coefficients for the constitutive equations need to be measured through various experiments, for example, the micropipette aspiration method. Well-known ES model was developed through the detail measurements by themselves. They expressed the strain energy function, which was fitted to the result of deformation measurements for human RBCs. Therefore, ES model can express the nonlinear elastic characteristics of RBC, although this model simplifies the complicated biomembrane structure, which has the so-called cytoskeleton structure below the lipid bilayer, as a hyperelastic membrane. On the other hand, lipid bilayer itself behaves with the fluidity in the tangential direction on the surface, which is an important feature of biomembrane called lateral diffusion.

For the numerical simulations of RBC, there are several ways to compute, using the schemes for fluid-structure interaction. For example, Pozrkidis [9] and Barthes-Biesel et al. [10] used the boundary element method (BEM), in which they solve the boundary integral equation for Stokes flow to simulate the deformation of vesicles. Eggleton and Popel [8] used the immersed boundary method (IBM) originally developed by Peskin [11]. From the viewpoint of numerical accuracy, BEM for Stokes flow will be better than the IBM as far as the simulation is conducted for small scale capillary vessels. But, considering the situation that the inertia effect of the flow field plays a certain role, IBM with Navier–Stokes equation will have some advantages. In both methods, the membrane is discretized into a network of surface elements defined by a collection of surface nodes. And tracking these nodes in Lagrangian way at each time step, a large deformation of RBC is simulated. Here, considering the future application for more complicated behavior, IBM was employed to simulate the large deformation of vesicles. Besides the IBM, the arbitrary Lagrangian-Eulerian (ALE) method with a moving mesh is also widely adopted in the simulations for the fluid-structure interaction problems [12,13]. The difference is as follows. In an ALE approach, the mesh is moved with an arbitrary velocity on which the flow field is solved. In the IBM, two separated grid systems is employed. It is on the stationary grids that the flow field is solved and on the unstructured surface grids that the movement and deformation of the membrane are tracked in a Lagrangian way. Thereafter, multiple RBCs in a complicated flow geometry would be easily treated with the IBM, although it could be less accurate than ALE method especially for the high Reynolds number cases.

In this paper, we numerically investigate the deformation of a vesicle in a linear shear flow using the immersed boundary method. First, two-dimensional simulation of a liposome tank-treading motion in a linear shear flow is investigated. Then, three-dimensional simulations of a liposome and a red blood cell in a linear shear flows are discussed with the emphasis on the difference of membrane models.

## 2 Membrane Models

Here, we explain the membrane models and the numerical method for the present simulation. There are two types of modeling for the lipid bilayer biomembrane. A liposome, which only has lipid bilayer structure, is modeled to have a two-dimensional incompressible fluid membrane due to the fluidity of lipid molecules on the membrane. On the other hand, red blood cell is often modeled to have a hyperelastic membrane due to their stiff cytoskeleton structure. In this study, the behaviors of vesicles, which have different types of membranes mentioned above, are discussed under the presence of shear flow.

**2.1 Liposome Model.** We first discuss the liposome membrane model, which has a simpler structure from molecular point of view. Here, we used the well-known Helfrich membrane model [14]. In the case of Helfrich model, membrane energy is given as follows, considering that the membrane shape is influenced by the bending rigidity.

$$E = \frac{1}{2}k_b \int (c_1 + c_2 - c_0)^2 dA + k_G \int c_1 c_2 dA + \lambda \int dA + \Delta P \int dV \tag{1}$$

where $k_b$ is the bending rigidity, $c_1$ and $c_2$ are the principal curvatures of the membrane, $c_0$ is the spontaneous curvature, and $dA$ and $dV$ are the small elements of surface and volume, respectively. Under this description, $\lambda$ and $\Delta P$ are the Lagrangian multipliers to keep the total surface area constant and the volume constant, respectively. $k_G \int c_1 c_2 dA$ term does not change as far as the topological change of a vesicle does not occur and is not considered in the present study. Using the expression given in Eq. (1), the force acting on a membrane is expressed by a functional derivative as

$$\mathbf{f} = -\frac{\delta E}{\delta \mathbf{r}} \tag{2}$$

**2.2 Red Blood Cell Model.** Next, the membrane model of RBC is discussed. RBC membrane is composed mainly by two parts: the lipid bilayer and a cytoskeleton of proteins. The membrane exhibits an elastic response to surface deformation and bending rigidity. Following Pozrikidis [9], we assume the RBC membrane to be a thin hyperelastic shell and express the jump in hydrodynamics traction across the RBC membrane as

$$\mathbf{f} = (\mathbf{P} \cdot \nabla) \cdot (\mathbf{T} + \mathbf{qn}) \tag{3}$$

Here, $\mathbf{P}$ is the tangential projection operator; $\mathbf{T}$ is the in-plane surface stress tensor. Here, $\mathbf{q}$ is the vector for the transverse shear stress and $\mathbf{n}$ is the outward unit normal vector. In the present study, the model of Skalak et al. [15] is employed for the surface stress tensor, which proposes the following expression for the membrane model of a RBC,

$$\mathbf{T} = \frac{B}{2\lambda_1\lambda_2}(\lambda_1^2 + \lambda_2^2 - 1)\mathbf{V^2} + \frac{\lambda_1\lambda_2}{2}\{C(\lambda_1^2\lambda_2^2 - 1) - B\}\mathbf{P} \quad (4)$$

where $\lambda_1$ and $\lambda_2$ are the principal strains, $\mathbf{V^2}$ is the positive-definite left-hand Cauchy–Green deformation tensor. $B$ and $C$ are the physical constants estimated to be on the order of $B=0.005$ dyn/cm and $C=100$ dyn/cm by Skalak et al. [15]. The first term on the right-hand side contributes to an anisotropic surface stress, while the second term contributes to an isotropic surface stress. The large value of $C$ gives the membrane nearly incompressible, that is, local area preserving.

**2.3 Numerical Method.** Peskin's immersed boundary method [11] is used for the simulation of deformable vesicles. Incompressible flow is assumed and the following governing equations are solved.

In continuity equation,

$$\nabla \cdot \mathbf{U} = 0 \quad (5)$$

In momentum equation,

$$\frac{D\rho\mathbf{U}}{Dt} = -\nabla p + \nabla \cdot \mu(\nabla\mathbf{U} + {}^t\nabla\mathbf{U}) + \int \mathbf{f}\delta(\mathbf{x} - \mathbf{X})dA \quad (6)$$

where $\mathbf{f}$ is the force acting on the membrane, which is calculated from the membrane model given in Eqs. (1)–(4) and $\delta(\mathbf{x}-\mathbf{X})$ represents Dirac's delta function where the membrane force acts on $\mathbf{x}=\mathbf{X}$. As is shown in Eq. (6), the force acting on the membrane is included in the momentum equation in this method. Since the direct use of the singular function $\delta(\mathbf{x}-\mathbf{X})$ gives a numerical instability, a smoothed delta function $D(\mathbf{x})$ given below is used in the discretized domain.

$$D(\mathbf{x}) = \begin{cases} (4h)^{-n}\prod_1^n \left(1 + \cos\frac{\pi}{2h}x_i\right) & (|x_i| < 2h) \\ 0 \quad \text{otherwise} \end{cases} \quad (7)$$

where $h$ is the grid size and $n$ indicates the spatial dimensions. The continuity equation (5) is coupled with the momentum equation (6) and the flow field is solved using SMAC algorithm. For the discretization, the second order central difference scheme is used for both the convective and viscous terms. Second order Adams–Bashforth is used for the time integration.

In the present study, the simulation was set up under the following conditions. The length scale of the flow field was given as an equivalent radius of the vesicles, which is defined as $a=(3V/4\pi)^{1/3}$. $a=2.8$ $\mu$m was employed here. The size of the computational domain was set as $8a\times 8a\times 8a$, and the number of grid points was $80\times 80\times 80$. A linear shear rate $\gamma$ was given in $y$-direction. The initial velocity field was set as $(\gamma y, 0, 0)$. In this study, the important Reynolds number is a shear Reynolds number for a vesicle, which is defined as $\mathrm{Re}=(2a)^2\gamma/\nu$. This Reynolds number is much smaller than unity and it is $O(10^{-5})$ in the present simulations. The time scale is given by the inverse of shear rates $1/\gamma$, and the time step for the present simulations is given as $10^{-4}$ of this time scale.

In general, the error of mass conservation of each separated region causes inaccurate numerical solutions in fluid-structure interaction problems. In the present methods, there is a small numerical error accumulating when the membrane surface is tracked in a Lagrangian way. This gives the mass (volume) conservation error. We have checked this error during the whole simulation period. It was found that the volume change in the present simulations is always smaller than 2%, mostly less than 1%. So, we concluded that the effect of this error is small enough to discuss the effect of membrane model differences.

For dimensionless numbers, swelling ratio is one of the most important. In a three-dimensional case, it is defined as

$$S_{w,3D} = \frac{V}{\frac{4\pi}{3}\left(\sqrt{\frac{S}{4\pi}}\right)^3} \quad (8)$$

where $V$ is the volume inside a vesicle, and $S$ is the surface area that the vesicle holds. The swelling ratio describes the ratio between the internal volume of a vesicle and that of a sphere, which has the same surface area. For a healthy red blood cell with the biconcave discoid shape formulated by Evans and Fung [16], as an example, the swelling ratio equals to 0.64.

With a similar idea, the swelling ratio in two-dimensional cases is defined as

$$S_{w,2D} = \frac{S}{4\pi\left(\frac{\ell}{2\pi}\right)^2} \quad (9)$$

where $S$ is the surface area, and $\ell$ is the perimeter length of the area. It takes the value of 1.0 when the area is circle.

The viscosities between the surrounding plasma and the components inside RBCs are usually different. The viscosity ratio between the inside and the outside of the RBC, defined as $r=\mu_{\mathrm{in}}/\mu_{\mathrm{out}}$ can be larger than 5.0. Both the swelling ratio $S_W$ and the viscosity ratio $r$ are important parameters to decide the tank-treading and tumbling behavior.

In the present work, the dimensionless parameters, $C_B$, $C_C$, and $C_b$ are also introduced. $C_B$, $C_C$, and $C_b$ are defined as follows:

$$C_B = \mu_{\mathrm{out}}\gamma a/B, \quad C_C = \mu_{\mathrm{out}}\gamma a/C, \quad C_b = \mu_{\mathrm{out}}\gamma a^3/k_b \quad (10)$$

These parameters indicate ratios of the shear stress in the flow to that in the membrane, the surface area expansion stress, and the bending stress, respectively. In Eq. (10), $B$ is a shear stress coefficient, $C$ is an expansion coefficient of the surface area, and $k_b$ is a bending rigidity. In the present simulations, $B$ takes the same value as that proposed by Chien et al. [17], which gives $B=1.7\times 10^{-3}$ dyn/cm. The bending rigidity $k_b$ is given from Evans and Skalak's work [7], in which $k_b=1.8\times 10^{-12}$ dyn cm. As for an expansion coefficient, Skalak et al. [15] proposed the value of $C=100$ dyn/cm. Since this value of $C$ is too large, the time step for the simulation has to be unreasonably small. And for the same reason, more instability is induced during the tank-treading motion when the membrane is rotating. To avoid this situation, a much smaller value of $C$, $C=0.01$ dyn/cm, is used in the present simulation. It is confirmed that even using this small number of $C$, the total surface area difference during the simulation is less than 1%.

For the boundary conditions, the periodic boundaries are used in $x$- and $z$-directions, and the moving wall conditions are imposed in $y$-direction. The linear shear profile of $(\gamma y, 0, 0)$ is also given inside the vesicle as an initial condition. The flow configuration and the initial setup for a RBC system are shown in Fig. 2.

## 3 Results and Discussion

**3.1 2D Simulation of a Liposome.** In this section, we discuss mainly the tank-treading motion and give the quantitative comparisons of the present results with those of other numerical and experimental results.

First, we discuss the two-dimensional case for the tank-treading motion of a liposome. There is a previous study done by Beaucourt et al. [3]. They conducted two-dimensional numerical simulations on the tank-treading motion of a liposome in a linear shear flow. In the present study, we start our discussion through the comparison with their results.

In the present simulations, a liposome is initially set in the computational domain without the presence of shear flow. In this situation, the liposome takes the equilibrium ellipsoidal-like shape after a certain interval of the simulation. This ellipsoidal-like shape is horizontally set in the domain; long-axis is parallel to the

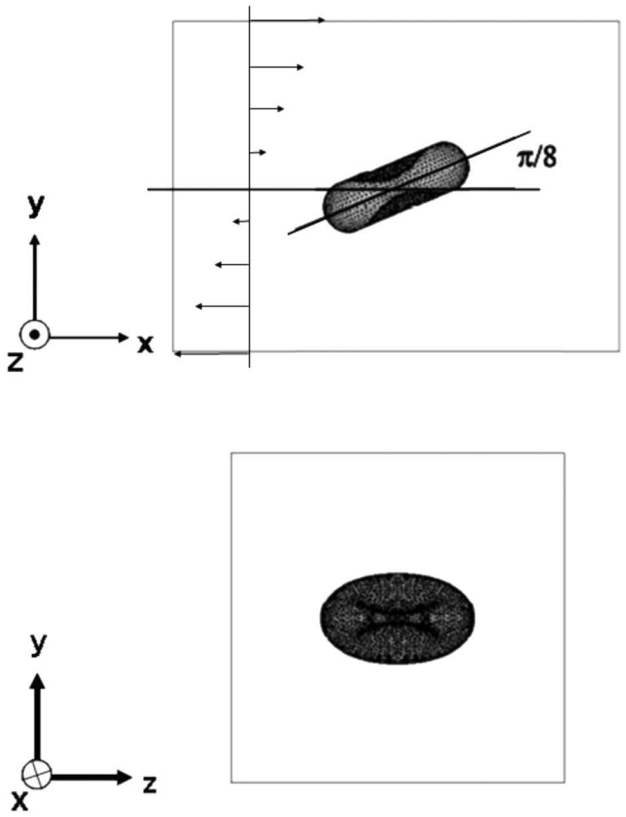



**Fig. 2 The initial configuration of a RBC in a linear shear flow**

$x$-axis. Then, a linear shear is introduced to have a tank-treading type motion. All the simulations for a deformed vesicle in a linear shear flow have been conducted from this initial condition.

Figure 3 indicates the dependence of inclination angle $\theta$ on the viscosity ratio ($r=\mu_{in}/\mu_{out}$). It is shown that the increase in viscosity ratio gives the decrease in inclination angle. A comparison between our results and the results of Beaucourt et al. [3] is also shown in the figure. Although their membrane model is similar to ours, their numerical method is different from ours and the constraint to satisfy the area preserving condition of membrane surface is also different. Even with these differences, it is interesting to see that both results have shown the good agreements. Hence, it

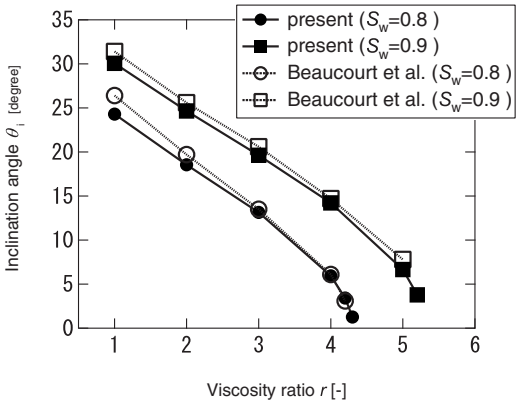



**Fig. 3 Dependence of inclination angle $\theta$ of 2D tank-treading motion on viscosity ratio**

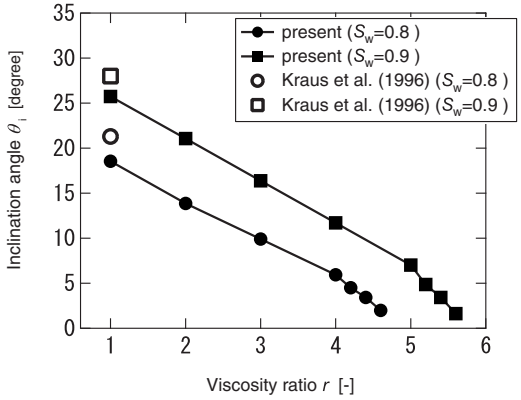



**Fig. 4 Dependence of inclination angle $\theta_i$ of 3D tank-treading motion on viscosity ratio**

is said that the present numerical method worked well and the present membrane model can properly reproduce the solution for the tank-treading motion of a vesicle.

**3.2 3D Simulation of a Liposome.** Next, we extend this method for the three-dimensional simulations. As is the same as two-dimensional cases, the initial shape is given by the solution of an equilibrium shape without shear. The equilibrium shape became ellipsoidal sticklike one, which is different from the well-known biconcave shape of the RBC. The difference comes from the membrane model. The present liposome model given by Eq. (1) has a minimum energy equilibrium shape as an ellipsoidal sticklike one.

Using this initial shape, three-dimensional simulation was conducted for the deformable liposome in a linear shear flow. The obtained results for the tank-treading motion are shown in Fig. 4. The results by Kraus et al. [4] for the viscosity ratio of 1.0 are also shown for the comparison. Although our numerical results show the slightly smaller value than their results, reasonable agreement is achieved. It should be mentioned that our membrane model has a geometric constraint of the total surface area constant, while Kraus et al. [4] gave the local surface area nearly kept constant.

In Fig. 4, it is shown that the inclination angle $\theta$ decreases with the increase in viscosity ratio. This tendency is exactly the same as 2D case. However, there is a quantitative difference for this inclination angle between 2D and 3D cases under the same swelling and viscosity ratios. To compare this difference in more details, the dependence of the inclination angle on the swelling ratio is shown in Fig. 5. The results are shown for both 2D and 3D cases. It is illustrated that a 3D liposome shows the lower inclination angle for the same swelling ratio. Since 2D and 3D are not geometrically the same and the meaning of the swelling ratio is different as explained for Eqs. (8) and (9), it is not strange to have

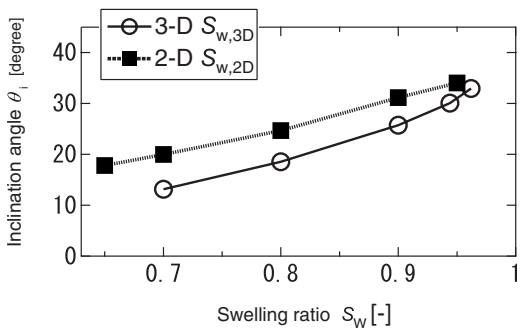



**Fig. 5 Dependence of inclination angle $\theta_i$ of tank-treading motion on swelling ratio (comparison of 2D and 3D results)**

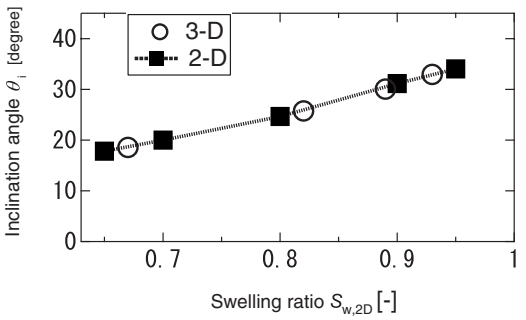



**Fig. 6 Dependence of inclination angle $\theta_t$ of tank-treading motion on swelling ratio (rearrangement of 2D and 3D results)**

quantitatively different results in 2D and 3D. However, it is very interesting to see that they show good agreements if the cross section of the 3D simulation was considered as 2D results in the symmetric plane of $z=0$ (Fig. 6). That is, if the 2D swelling ratio is obtained from the cross section area in the symmetric plane of the 3D simulation, this 2D discussion using the 3D results show good agreement with the results of the 2D simulations.

Related to this tank-treading motion, the experiment was also conducted in our group [18]. Liposomes were produced by the so-called gentle hydration method and they are introduced in a linear shear flow. Tank-treading angle was measured using the high speed camera and microscopy. The comparisons of the tank-treading shapes between the photos taken in the experiments and the present numerical simulations are shown in Fig. 7. It is illustrated that good agreements are attained for both the shapes and inclination angles. More quantitative comparison for the inclination angles among the present numerical simulation, our experiment [18], experiment by Abkarian et al. [19], theory by Keller and Skalak [20], and the simulation by Kraus et al. [4] are given in Fig. 8. It illustrates that all the data show quantitatively good agreement. It is noted that the theory by Keller and Skalak [20] assumed the ellipsoidal shape and it is not the cases of our simulations or the simulations of Kraus et al. [4]. This indicates that the swelling ratio is more dominant factor than a small difference of membrane model is.

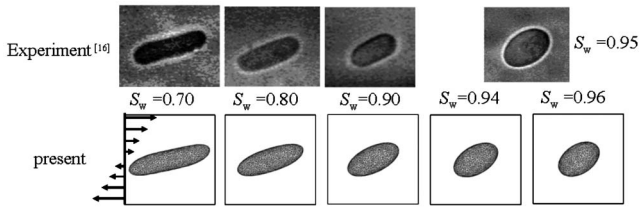



**Fig. 7 Comparison of the tank-treading shapes between experiments and the present simulations**

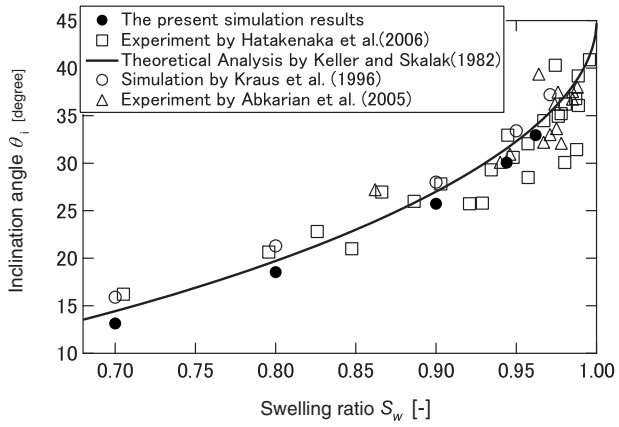



**Fig. 8 Comparison of the inclination angle with other results**

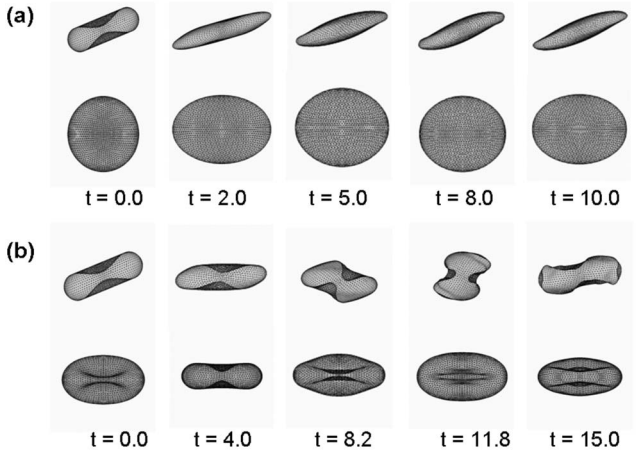



**Fig. 9 The motion of a RBC in a linear shear flow. (*a*) tank-treading motion: $\mu_{in}/\mu_{out}=0.35$, $\gamma=950$ 1/s, and $B=4.2\times10^{-3}$ dyn/cm. (*b*) tumbling motion: $\mu_{in}/\mu_{out}=6.2$, $\gamma=950$ 1/s, and $B=4.2\times10^{-3}$ dyn/cm.**

**3.3 3D Simulation of a Red Blood Cell and the Comparison With Liposome.** Here, we discuss the dependence of different membrane models. We use a hyperelastic membrane model for a RBC. In the case of RBC simulation, the initial shape is given as a biconcave discoid, which is formulated by Evans and Fung [16]. The initial condition of RBC in a linear shear flow has been illustrated in Fig. 2, where the initial inclination angle is set at $\pi/8$.

Figures 9(*a*) and 9(*b*), respectively, illustrate the tank-treading and the tumbling motion of RBC. In the case of Fig. 9(*b*), the ratio of the viscosity between the inside and outside of the RBC is around 6.2, and the swelling ratio is around 0.64, which are in the actual range of the RBC properties. As shown in the figure, when the simulation starts, the RBC starts tumbling in clockwise direction. Compared with the tumbling motion of "liposome" in the same simulation condition, in which liposomes are extended in the shear and the dynamic shape of rod and round plate in two different directions are observed, the tumbling motion of the RBC shows a different transient shape. This is due to the difference of the membrane resistance to the flow shear stress. These instantaneous shapes are similar to the one observed in the experiment by Fischer and Schonbein [21].

To investigate the effect of viscosity ratio, the viscosity ratio between the inside and outside of the RBC was set to 0.35, which does not occur in real situations. Under this condition, the tank-treading motion of RBC occurred, as shown in Fig. 9(*a*). The upper figures show the view of RBC in $x$-$z$ plane, which is different from the lower figures (in $y$-$z$ plane). As shown in the figure, the extension and reduction of the length of the shape are observed. The angle between the $x$ coordinate and the long axis of the RBC keeps almost a steady value, and the tank-treading motion with the movement of membrane between two fluids with different viscosities is observed. Compared with the deformation of a liposome in a tank-treading motion, the liposome keeps extension before the rod shape is obtained, while for the deformation of the RBC, with the change in elliptic shape, the length of the rod of the RBC does not change obviously.

At last, we discuss the comparison of tank-treading angle between the liposomes and vesicles with hyperelastic RBC membrane. The results are shown in Fig. 10. It is noted that the swelling ratio of 0.8 is rather large as a RBC, so this is a kind of vesicles that has a cell membrane structure like RBC. Under the same swelling ratio and the viscosity ratio, the vesicles with RBC

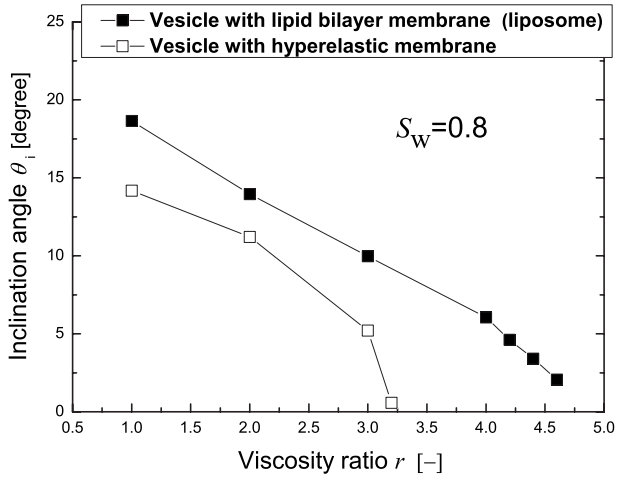



**Fig. 10 The effect of membrane model to the inclination angle of tank-treading motion**

membrane models show the smaller inclination angle than those with Helfrich [14] fluid membrane models: liposomes. The vesicles with RBC membrane models also show an earlier transition from tank-treading. That is, the hyperelastic membrane model for a RBC shows a stiffer behavior than that for a liposome in a wide parameter range.

## 4 Conclusion

In this paper, the motion of a deformable vesicle in a linear shear flow was discussed. Tank-treading motions were mainly discussed for liposome. Then, the effect of membrane model, 2D fluid membrane for liposome and hyperelastic membrane for RBC, were discussed. The following results were obtained.

(1) The present 2D results for the tank-treading motion show good agreement with those by Beaucout et al. [3].
(2) 3D tank-treading motion shows qualitatively the same tendency as that of 2D. Especially, if the 3D results are analyzed such as the 2D data at the cross section in a symmetric plane, the inclination angle of the 3D simulations show the excellent agreement with those of the 2D.
(3) The hyperelastic membrane model for a RBC shows a stiffer behavior than that for a liposome. Under the same swelling ratio and the viscosity ratio, the vesicles with RBC membrane models show the smaller inclination angle than those with Helfrich fluid membrane models. The vesicles with RBC membrane models also show an earlier transition from tank-treading to tumbling due to their stiff membrane characteristic.

## Acknowledgment

This research was supported by Research and Development of the Next-Generation Integrated Simulation of Living Matter, a part of the Development and Use of the Next-Generation Supercomputer Project of the Ministry of Education, Culture, Sports, Science and Technology (MEXT).

## Nomenclature

$a$ = volumetric equivalent radius
$B$ = shear stress coefficient
$C$ = expansion coefficient of membrane surface area
$c_0$ = spontaneous curvature
$c_1, c_2$ = principal curvature of membrane
$dA$ = infinitesimal surface element
$dV$ = infinitesimal volume element
$k_b$ = bending rigidity
$\mathbf{n}$ = outward unit normal vector
$p$ = pressure
$\mathbf{P}$ = tangential projection operator
$\mathbf{q}$ = vector for the transverse shear stress
$r$ = viscosity ratio
$S_w$ = swelling ratio
$\mathbf{T}$ = in-plane surface stress tensor
$\mathbf{U}$ = flow velocity field

### Greek Symbols

$\gamma$ = shear rate
$\lambda_1, \lambda_2$ = principal strain
$\mu$ = viscosity
$\nu$ = kinematic viscosity of external fluid
$\theta$ = inclination angle of tank-treading motion

## References

[1] Sugii, T., Takagi, S., and Matsumoto, Y., 2005, "A Molecular-Dynamics Study of Lipid Bilayers: Effects of the Hydrocarbon Chain Length on Permeability," J. Chem. Phys., **123**, p. 184714.
[2] Sugii, T., Takagi, S., and Matsumoto, Y., 2007, "A Meso-Scale Analysis of Lipid Bilayers With the Dissipative Particle Dynamics Method: Thermally Fluctuating Interfaces," Int. J. Numer. Methods Fluids, **54**, pp. 831–840.
[3] Beaucourt, J., Rioual, F., Seon, T., Biben, T., and Misbah, C., 2004, "Steady to Unsteady Dynamics of a Vesicle in a Flow," Phys. Rev. E, **69**, p. 011906.
[4] Kraus, M., Wintz, W., and Seifert, U., 1996, "Fluid Vesicles in Shear Flow," Phys. Rev. Lett., **77**(17), pp. 3685–3688.
[5] Boryczko, K., Dzwinel, W., and Yuen, D. A., 2003, "Dynamical Clustering of Red Blood Cells in Capillary Vessels," J. Mol. Model., **9**, pp. 16–33.
[6] Tanaka, M., and Koshizuka, S., 2007, "Simulation of Red Blood Cell Deformation Using a Particle Method," Nagare, **26**, pp. 49–55.
[7] Evans, E. A., and Skalak, R., 1980, *Mechanics and Thermodynamics of Biomembranes*, CRC, Bota Raton, FL.
[8] Eggleton, C. D., and Popel, S., 1998, "Large Deformation of Red Blood Cell Ghosts in a Simple Shear Flow," Phys. Fluids, **10**, pp. 1834–1845.
[9] Pozrikidis, C., 2001, "Effect of Membrane Bending Stiffness on the Deformation of Capsules in Simple Shear Flow," J. Fluid Mech., **440**, pp. 269–291.
[10] Barthes-Biesel, D., Diaz, A., and Dhenin, E., 2002, "Effect of Constitutive Laws for Two-Dimensional Membranes on Flow-Induced Capsule Deformation," J. Fluid Mech., **460**, pp. 211–222.
[11] Peskin, C., 1977, "Numerical Analysis of Blood Flows in the Heart," J. Comput. Phys., **25**, pp. 220–252.
[12] Tezduyar, T. E., Sathe, S., Keedy, R., and Stein, K., 2006, "Space-Time Finite Element Techniques for Computation of Fluid-Structure Interactions," Comput. Methods Appl. Mech. Eng., **195**, pp. 2002–2027.
[13] Khurram, R., and Masud, A., 2006, "A Multiscale/Stabilized Formulation of the Incompressible Navier–Stokes equations for Moving Boundary Flows and Fluid-Structure Interaction," Comput. Mech., **38**(4–5), pp. 403–416.
[14] Helfrich, W., 1973, "Elastic Properties of Lipid Bilayers: Theory and Possible Experiments," Z. Naturforsch [C], **28**, pp. 693–703.
[15] Skalak, R., Tozeren, A., Zarda, R., and Chien, S., 1973, "Strain Energy Function of Red Blood Cell Membranes," Biophys. J., **13**, pp. 245–264.
[16] Evans, E., and Fung, Y., 1972, "Improved Measurement of the Erythrocyte Geometry," Microvasc. Res., **4**, pp. 335–347.
[17] Chien, S., Sung, K., Skalak, R., Usami, S., and Tozeren, A., 1978, "Theoretical and Experimental Studies on Viscoelastic Properties of Erythrocyte Membrane," Biophys. J., **24**, pp. 463–487.
[18] Hatakenaka, R., Takagi, S., and Matsumoto, Y., 2008, "The Behavior of a Lipid Bilayer Vesicle in a Simple Shear Flow (1st Report, Validation of the Relationship Between Inclination Angle and Swelling Ratio)," Trans. Jpn. Soc. Mech. Eng., Ser. B, **74**, pp. 530–535.
[19] Abkarian, M., Lartigue, C., and Viallat, A., 2002, "Tank Treading and Unbinding of Deformable Vesicles in Shear Flow: Determination of the Lift Force," Phys. Rev. Lett., **88**(6), p. 068103.
[20] Keller, S., and Skalak, R., 1982, "Motion of a Tank-Treading Ellipsoidal Particle in a Shear Flow," J. Fluid Mech., **120**, pp. 27–47.
[21] Fischer, T., and Schonbein, H. S., 1977, "Tank Tread Motion of Red Cell Membranes in Viscometric Flow: Behavior of Intracellular and Extracellular Markers (With Film)," Blood Cells, **3**, pp. 351–365.

**Lucia Catabriga**
Department of Computer Science,
Federal University of Espírito Santo,
Avenida Fernando Ferrari 514,
Vitória, ES 29075-910, Brazil
e-mail: luciac@inf.ufes.br

**Denis A. F. de Souza**
Laboratory for Computational Methods in Engineering (LAMCE),
Federal University of Rio de Janeiro,
P.O. Box 68506,
Rio de Janeiro,
RJ 21945-970, Brazil
e-mail: denis@lamce.coppe.ufrj.br

**Alvaro L. G. A. Coutinho**
Department of Civil Engineering-COPPE and Center for Parallel Computations,
Federal University of Rio de Janeiro,
P.O. Box 68506,
Rio de Janeiro, RJ 21945-970, Brazil
e-mail: alvaro@nacad.ufrj.br

**Tayfun E. Tezduyar**
Mechanical Engineering,
Rice University,
MS 321, 6100 Main Street,
Houston, TX 77005
e-mail: tezduyar@rice.edu

# Three-Dimensional Edge-Based SUPG Computation of Inviscid Compressible Flows With $YZ\beta$ Shock-Capturing

*The streamline-upwind/Petrov–Galerkin (SUPG) formulation of compressible flows based on conservation variables, supplemented with shock-capturing, has been successfully used over a quarter of a century. In this paper, for inviscid compressible flows, the $YZ\beta$ shock-capturing parameter, which was developed recently and is based on conservation variables only, is compared with an earlier parameter derived based on the entropy variables. Our studies include comparing, in the context of these two versions of the SUPG formulation, computational efficiency of the element- and edge-based data structures in iterative computation of compressible flows. Tests include 1D, 2D, and 3D examples.* [DOI: 10.1115/1.3062968]

*Keywords: inviscid compressible flow, SUPG formulation, stabilization parameter, $YZ\beta$ shock-capturing, edge-based data structure*

## 1 Introduction

The streamline-upwind/Petrov–Galerkin (SUPG) formulation of compressible flows is widely used in finite element flow computations. The formulation was introduced, within a couple of years following the introduction of the SUPG formulation of incompressible flows [1,2], in a NASA Technical Report [3]. A concise version of the technical report was published as an AIAA paper [4] and a more thorough version with additional examples as a journal article [5], which has been more widely available than the first two versions. The SUPG formulation of compressible flows introduced in Refs. [3–5] was in the context of conservation variables and without a shock-capturing term. Following Refs. [3–5], the SUPG formulation of compressible flows was recast in entropy variables and supplemented with a shock-capturing term [6]. It was shown first in Ref. [7], and later in Ref. [8], that the SUPG formulation introduced in Refs. [3–5], when supplemented with a similar shock-capturing term, is very comparable in accuracy to the one that was recast in entropy variables.

The SUPG formulation of compressible flows, just like the SUPG formulation of incompressible flows and most other stabilized formulations following the two, involves a stabilization parameter that is mostly known as $\tau$. This parameter represents a measure of the local length scale (also known as "element length") and other parameters such as the element Reynolds and Courant numbers. Various $\tau$ definitions were proposed starting with those in Refs. [1–5], followed by the one introduced in Ref. [9], and those proposed in the subsequently reported SUPG methods. Here we will call the SUPG formulation introduced in Refs. [3–5] for compressible flows $(\text{SUPG})_{82}$, and the set of $\tau$s introduced in conjunction with that formulation $\tau_{82}$. The stabilized formulation introduced in Ref. [9] for advection-diffusion reaction equations included a shock-capturing term and a $\tau$ definition that takes into account the interaction between the shock-capturing and SUPG terms. The $\tau$ used in Ref. [7] with $(\text{SUPG})_{82}$ is a slightly modified version of $\tau_{82}$. A shock-capturing parameter, which was derived from its counterpart in the entropy variables and which we will call here $\delta_{91}$, was embedded in the shock-capturing term used in Ref. [7]. Subsequent minor modifications of $\tau_{82}$ took into account the interaction between the shock-capturing and the $(\text{SUPG})_{82}$ terms in a fashion similar to how it was done in Ref. [9] for advection-diffusion reaction equations. All these slightly modified versions of $\tau_{82}$ have always been used with the same $\delta_{91}$, and we will categorize them here all under the label $\tau_{82\text{-MOD}}$.

New ways of computing the $\tau$s based on the element matrices and vectors were introduced in Ref. [10] in the context of the advection-diffusion equation and the Navier–Stokes equations of incompressible flows. These new definitions are expressed in terms of the ratios of the norms of the matrices or vectors. In Refs. [11,12], the $\tau$ definitions based on the element matrices were used in conjunction with the $(\text{SUPG})_{82}$ formulation supplemented with the shock-capturing term involving $\delta_{91}$. In Ref. [13], these definitions were extended to the edge-based implementation, which was introduced in Ref. [14]. The edge-based implementation is computationally more efficient than the element-based implementation.

The $YZ\beta$ shock-capturing was introduced in Refs. [15–17]. It is based on using a simple residual-based shock-capturing parameter and is less costly to compute than $\delta_{91}$. It has options for smoother or sharper computed shocks. It was tested in Refs. [18–20]. Those test computations showed that the $YZ\beta$ shock-capturing parameters are not only much simpler than $\delta_{91}$ but also superior in

Contributed by the Applied Mechanics Division of ASME for publication in the JOURNAL OF APPLIED MECHANICS. Manuscript received July 8, 2008; final manuscript received August 28, 2008; published January 15, 2009. Review conducted by Arif Masud.

accuracy. In Ref. [21], the $YZ\beta$ shock-capturing was used in combination with the variable subgrid scale (V-SGS) method, which was introduced in Ref. [22] and was formulated for compressible flows in conservation variables in Ref. [23].

In this paper, we extend the SUPG formulation with $YZ\beta$ shock-capturing to the edge-based implementation. We carry out test computations in 1D, 2D, and 3D and compare the performance of the $YZ\beta$ shock-capturing parameter to the performance of $\delta_{91}$. We also compare, in the context of these two versions of the SUPG formulation, the computational efficiency of the element- and edge-based data structures used in iterative computation of compressible flow problems. We describe the governing equations in Sec. 2 and numerical formulation in Sec. 3, and present the numerical examples in Sec. 4. The concluding remarks are given in Sec. 5.

## 2 Governing Equations

The 3D equations of inviscid compressible flows can be written as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} = 0 \quad \text{in} \quad \Omega \times [0, T_{\max}] \tag{1}$$

where $\Omega \subset \mathbb{R}^3$ and $t \in [0, T_{\max}]$. The vector of conservation variables $\mathbf{U}$ and the vector of inviscid fluxes $\mathbf{F}_i$ are given as

$$\mathbf{U} = \rho \begin{Bmatrix} 1 \\ u_1 \\ u_2 \\ u_3 \\ e \end{Bmatrix}, \quad \mathbf{F}_i = \rho u_i \begin{Bmatrix} 1 \\ u_1 \\ u_2 \\ u_3 \\ e \end{Bmatrix} + p \begin{Bmatrix} 0 \\ \delta_{1i} \\ \delta_{2i} \\ \delta_{3i} \\ u_i \end{Bmatrix} \tag{2}$$

Here $\rho$ is the density, $\mathbf{u} = [u_1, u_2, u_3]^T$ is the velocity vector, $e$ is the total energy density, $p$ is the pressure, and $\delta_{ij}$ is the Kronecker delta. The equation of state used here corresponds to the ideal gas assumption. Alternatively, Eq. (1) can be written as

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}_i \frac{\partial \mathbf{U}}{\partial x_i} = \mathbf{0} \ \text{ in } \ \Omega \times [0, T_{\max}], \quad \mathbf{A}_i = \frac{\partial \mathbf{F}_i}{\partial \mathbf{U}} \tag{3}$$

Appropriate sets of boundary and initial conditions are assumed to accompany Eq. (3).

## 3 Numerical Formulation

**3.1 Finite Element Formulation.** We assume that we have constructed some suitably defined finite-dimensional trial solution and test function spaces $\mathcal{S}^h$ and $\mathcal{V}^h$. Based on that, the SUPG formulation [4,5,7] can be written as follows: find $\mathbf{U}^h \in \mathcal{S}^h$ such that $\forall \mathbf{W}^h$:

$$\int_\Omega \mathbf{W}^h \cdot \left( \frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i} \right) d\Omega + \sum_{e=1}^{\text{nel}} \int_{\Omega^e} \boldsymbol{\tau} \left( \frac{\partial \mathbf{W}^h}{\partial x_k} \right) \cdot \mathbf{A}_k^h \left( \frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i} \right) d\Omega + \sum_{e=1}^{\text{nel}} \int_{\Omega^e} \nu_{\text{SHOC}} \left( \frac{\partial \mathbf{W}^h}{\partial x_i} \right) \cdot \left( \frac{\partial \mathbf{U}^h}{\partial x_i} \right) d\Omega = 0 \tag{4}$$

At each time step, the coupled nonlinear equations involved are solved with the predictor-multicorrector algorithm described for compressible flows in Ref. [5]. An iterative technique with nodal-block-diagonal preconditioner and GMRES update method [24] is employed for solving the linear equation system involved. The SUPG stabilization and shock-capturing parameters are denoted by $\boldsymbol{\tau}$ and $\nu_{\text{SHOC}}$. They will be discussed in Secs. 3.2–3.4.

**3.2 Stabilization Parameters.** In our test computations, we compare the performance of the new stabilization parameters to the performance of $\tau_{82\text{-MOD}}$, which we use in the form $\boldsymbol{\tau} = \tau_{82\text{-MOD}} \mathbf{I}$. Therefore, we first provide the definition of $\tau_{82\text{-MOD}}$, which is a product of an evolution process that started with Refs. [3–5], continued with Ref. [9], reached maturity in Ref. [7], and involved further adjustments in subsequent publications. The specific form of $\tau_{82\text{-MOD}}$ given here is from Ref. [25]:

$$\tau_{82\text{-MOD}} = \max(0, \tau_t + \zeta(\tau_a - \tau_\delta)) \tag{5}$$

$$\tau_t = \frac{2}{3(1 + 2\alpha Cr)} \tau_a, \quad \tau_a = \frac{h}{2u_{cc}}, \quad \tau_\delta = \frac{\delta_{91}}{(u_{cc})^2} \tag{6}$$

where $h$ is an element length defined as the cubic root of the element volume, and $\zeta$, $Cr$, and $u_{cc}$ are defined as follows:

$$\zeta = \frac{2\alpha Cr}{1 + 2\alpha Cr}, \quad Cr = \frac{u_{cc}\Delta t}{h}, \quad u_{cc} = c + \mathbf{u}^h \cdot \frac{\nabla \|\mathbf{U}^h\|}{\|\nabla \|\mathbf{U}^h\|\|} \tag{7}$$

Here $c$ is the acoustic speed, $\alpha$ is a parameter controlling the stability and accuracy of the time-marching algorithm (set as $\alpha = 0.5$ here), and $\Delta t$ is the time-step size.

New ways of calculating the stabilization parameters in the context of the $(\text{SUPG})_{82}$ were introduced in Refs. [15–17] and tested in Refs. [18–20]. Here we provide from Refs. [15–17] the version we use in our computations. For this purpose, we first define the unit vector $\mathbf{j} = \nabla \rho^h / \|\nabla \rho^h\|$. The stabilization parameter $\tau_{\text{UGN}}$ is defined from its components corresponding to the advection- and transient-dominated limits, $\tau_{\text{SUGN1}}$ and $\tau_{\text{SUGN2}}$. In computing $\tau_{\text{SUGN1}}$ for each component of the test vector-function $\mathbf{W}$, the stabilization parameters $\tau^\rho_{\text{SUGN1}}$, $\tau^\mu_{\text{SUGN1}}$, and $\tau^e_{\text{SUGN1}}$ (associated with $\rho$, $\rho\mathbf{u}$, and $\rho e$, respectively) are defined by the following expression:

$$\tau^\rho_{\text{SUGN1}} = \tau^\mu_{\text{SUGN1}} = \tau^e_{\text{SUGN1}} = \left( \sum_{a=1}^{n_{\text{en}}} |\mathbf{u}^h \cdot \nabla N_a| \right)^{-1} \tag{8}$$

In computing $\tau_{\text{SUGN2}}$, the parameters $\tau^\rho_{\text{SUGN2}}$, $\tau^\mu_{\text{SUGN2}}$, and $\tau^e_{\text{SUGN2}}$ are defined as follows:

$$\tau^\rho_{\text{SUGN2}} = \tau^\mu_{\text{SUGN2}} = \tau^e_{\text{SUGN2}} = \frac{\Delta t}{2} \tag{9}$$

The parameters $\tau^\rho_{\text{UGN}}$, $\tau^\mu_{\text{UGN}}$, and $\tau^e_{\text{UGN}}$ are calculated from their components by using the $r$-switch [10]:

$$\tau^\rho_{\text{UGN}} = \left( \frac{1}{(\tau^\rho_{\text{SUGN1}})^r} + \frac{1}{(\tau^\rho_{\text{SUGN2}})^r} \right)^{-1/r} \tag{10}$$

$$\tau^\mu_{\text{UGN}} = \left( \frac{1}{(\tau^\mu_{\text{SUGN2}})^r} + \frac{1}{(\tau^\mu_{\text{SUGN2}})^r} \right)^{-1/r} \tag{11}$$

$$\tau^e_{\text{UGN}} = \left( \frac{1}{(\tau^e_{\text{SUGN1}})^r} + \frac{1}{(\tau^e_{\text{SUGN2}}))^r} \right)^{-1/r} \tag{12}$$

Typically $r = 2$. Thus, the resulting diagonal stabilization-parameter matrix $\boldsymbol{\tau}_{\text{UGN}}$ is written as

$$\boldsymbol{\tau}_{\text{UGN}} = \begin{bmatrix} \tau^\rho_{\text{UGN}} & & & & & \\ & \tau^\mu_{\text{UGN}} & & & & \\ & & \tau^\mu_{\text{UGN}} & & & \\ & & & \tau^\mu_{\text{UGN}} & & \\ & & & & \tau^e_{\text{UGN}} & \end{bmatrix} \tag{13}$$

**3.3 Shock-Capturing Parameters.** We also compare the performance of the new shock-capturing parameters to the performance of $\delta_{91}$. For that reason, we first provide the definition of $\delta_{91}$ [7]:

$$\delta_{91} = \left\| \mathbf{A}_k^h \frac{\partial \mathbf{U}^h}{\partial x_k} \right\|_{\tilde{\mathbf{A}}_0^{-1}} \Bigg/ \left( \sum_{j=1}^{n_{sd}} \left\| \frac{\partial \xi_j}{\partial x_k} \frac{\partial \mathbf{U}^h}{\partial x_k} \right\|_{\tilde{\mathbf{A}}_0^{-1}}^2 \right)^{1/2} \tag{14}$$

where $\xi_j$s are the element coordinates, and $\tilde{\mathbf{A}}_0$ is the Jacobian of the transformation from the entropy variables to the conservation variables.

The $YZ\beta$ shock-capturing was introduced in Refs. [15–17] and tested in Refs. [18–20]. Here we provide from Refs. [15–17] the version we use in our computations:

$$\nu_{SHOC} = \|\mathbf{Y}^{-1}\mathbf{Z}\| \left( \sum_{a=1}^{n_{sd}} \left\| \mathbf{Y}^{-1} \frac{\partial \mathbf{U}^h}{\partial x_i} \right\|^2 \right)^{\beta/2-1} \|\mathbf{Y}^{-1}\mathbf{U}\|^{1-\beta} \left( \frac{h_{shoc}}{2} \right)^{\beta} \tag{15}$$

where $\mathbf{Y}$ is a diagonal scaling matrix constructed from the reference values of the components of $\mathbf{U}$:

$$\mathbf{Y} = \begin{bmatrix} (U_1)_{ref} & & & & \\ & (U_2)_{ref} & & & \\ & & (U_3)_{ref} & & \\ & & & (U_4)_{ref} & \\ & & & & (U_5)_{ref} \end{bmatrix} \tag{16}$$

$$\mathbf{Z} = \frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i} \tag{17}$$

$$h_{shoc} = 2 \left( \sum_{a=1}^{n_{en}} |\mathbf{j} \cdot \nabla N_a| \right)^{-1} \tag{18}$$

The parameter $\beta$ is set as $\beta=1$ for smoother shocks and $\beta=2$ for sharper shocks. The compromise between the $\beta=1$ and $\beta=2$ selections was defined in Refs. [15–17] as the following averaged expression for $\nu_{SHOC}$:

$$\nu_{SHOC} = \tfrac{1}{2}((\nu_{SHOC})_{\beta=1} + (\nu_{SHOC})_{\beta=2}) \tag{19}$$

**3.4 Edge-Based Solver.** Following the algebraic approach in Ref. [14], the element matrices can be disassembled into their edge contributions. For all elements sharing a given edge, one can add their terms and construct the edge matrix, which is a $10 \times 10$ nonsymmetric matrix. The edge matrix is smaller than the element matrix, which is $20 \times 20$, but the number of edges is always greater than the number of elements. The number of terms stored in the edge-based data structure requires less memory and fewer computations than the element-based data structure. Note, however, that different from element-based data structures, the edge-based data structures require the scatter and add of element contributions to the six edges. This is clearly an overhead, present also when you use other data structures, as for instance, sparse formats.

Further gains can be achieved when using a block-diagonal preconditioner, as in our case. There is no need to store the edge matrix block-diagonals but only the inverse of the global block-diagonal $\mathbf{B}_d$. This way of performing the matrix-vector products, either element or edge based, as presented in Ref. [26] for the scalar transport equation, can be used for the problem at hand here in the following manner:

$$\mathbf{A}\mathbf{x} = \mathbf{B}_d\mathbf{x} + \sum_{s=1}^{ned} (\mathbf{A}_{off}^s)x^s \tag{20}$$

$$\mathbf{B}_d^{-1}\mathbf{A}\mathbf{x} = \mathbf{B}_d^{-1}\mathbf{B}_d\mathbf{x} + \mathbf{B}_d^{-1}\sum_{s=1}^{ned} (\mathbf{A}_{off}^s)x^s \tag{21}$$

**Table 1 Computational requirements for the element- and edge-based data structures with tetrahedral meshes**

| | Storage | i.a. | Flops |
|---|---|---|---|
| Element | 300 nel (1650$N$) | 360 nel (3520$N$) | 600 nel (4200$N$) |
| Edge | 50 nel (350$N$) | 90 nel (630$N$) | 100 nel (550$N$) |

$$\mathbf{B}_d^{-1}\mathbf{A}\mathbf{x} = \mathbf{x} + \mathbf{B}_d^{-1}\sum_{s=1}^{ned} (\mathbf{A}_{off}^s)x^s \tag{22}$$

where "ned" is the number of edges, and $\mathbf{A}_{off}^s$ is the matrix of off-diagonal terms associated with edge $s$.

We note that the resulting nonsymmetric edge matrix has 100 terms to be stored, minus the block-diagonal terms, which are 50. Thus, there is only need to store half of the terms, since the inverted block-diagonals are required for preconditioning purposes. We also note that this gain is more significant for the edge-based data structure than it is for the element-based one; because for the element-based approach only 100 of the 400 terms are in the block-diagonal and do not need to be stored.

Table 1 shows, for computation of matrix-vector products with five degrees of freedom and element- and edge-based data structures for tetrahedral meshes with $N$ nodes, the storage requirements for the effective mass-matrix coefficients (storage), the costs of indirect addressing (i.a.), and the floating point operations (flops). According to the estimates given in Ref. [27] for tetrahedral meshes, nel=5.5$N$ and ned=7$N$. Table 1 demonstrates the superiority of the edge-based data structure over the element-based one, both in memory requirements and operation count. In many cases, the edge-based data structure does not present a good balance between floating point and i.a. operations. To improve this ratio, several alternatives to the basic edge-based approach were proposed in Ref. [28], which are based on reusing the already gathered data as much as possible. This idea, combined with node renumbering strategies, introduces further enhancements, as shown in Ref. [28]. The basic approach is used here because it is simpler and provides better computational performance for the problems computed.

## 4 Numerical Examples

All coefficients of the effective mass matrix and residual vector were calculated with the aid of a symbolic mathematical software, resulting in a code with one single loop over the elements to compute and assemble the edge-matrices and residual. The effective mass-matrix routine has around 6400 lines of code. All meshes are made of linear tetrahedra. The flops count for computing $\delta_{91}$ is 240, while for the $YZ\beta$ shock-capturing parameter ($\nu_{SHOC}$) it is just 160. In the computations presented, $\nu_{SHOC}$ is used in conjunction with the stabilization parameter given by Eq. (13), and $\delta_{91}$ is used with the stabilization parameter given by Eq. (5). All units are omitted due to the nondimensionalization of the variables involved.

**4.1 1D Shock Tube.** A shock tube problem is an essentially 1D flow discontinuity problem that provides a good test for compressible flows simulations. The domain is a cylindrical or rectangular tube, with a middle membrane barrier separating two initial gas states at different pressures and densities. The pressure and density can be high in one of the halves and low in the other. The rupture of the middle barrier at time $t=0$ allows the two halves to interact. The well-known Sod shock tube benchmark problem is

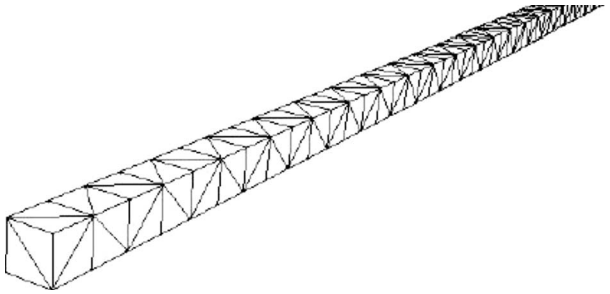


**Fig. 1 1D shock tube. Mesh.**

considered here. The solution contains simultaneously a shock wave, a contact discontinuity, and an expansion fan. A reference solution was obtained using a fine mesh with $1000 \times 1$ cells, each divided into five tetrahedra. This solution is in agreement with the analytical solution (see Ref. [29]). In our test computation, we use a mesh with $100 \times 1$ cells, which is shown in Fig. 1.

The initial condition consists of $\rho=1.0$, $u_1=0.0$, and $p=1.0$ on the left side ($0 \leq x_1 < 0.5$) and $\rho=0.125$, $u_1=0.0$, and $p=0.1$ on the right side ($0.5 \leq x_1 \leq 1.0$). The boundary conditions at $x_1=0$ and $x_1=1$ are set to the values at the corresponding halves of the domain. The velocity components $u_2$ and $u_3$ are set to zero. The time-step size is $10^{-3}$, and the simulation duration is 0.2. For the preconditioned GMRES solver, 30 vectors are used in the Krylov basis, with a maximum of 10 cycles and a tolerance of $10^{-9}$. Nonlinear tolerance is $10^{-3}$ and convergence is achieved at every time step. We tested two different options for the reference values used in Eq. (16). In the option denoted by $\nu_{\text{SHOC left}}$, we use the initial condition values for the left domain. In the option denoted by $\nu_{\text{SHOC previous}}$, we use the values from the previous iteration. We show the solutions in Fig. 2, together with the solution obtained with $\delta_{91}$. The $\delta_{91}$ solution is more dissipative and both $YZ\beta$ solutions capture the shocks better. The $\nu_{\text{SHOC left}}$ solution converges in four to five iterations for all time steps, but the convergence of the $\nu_{\text{SHOC previous}}$ solution stagnates. This is due to the additional nonlinearity introduced by the reference values used in computation of the shock-capturing parameter. In the subsequent test computations in this paper, we will use fixed reference values, typically those corresponding to the initial condition.

**4.2 2D Flow in a Channel With a Step.** The problem of a wind tunnel containing a step was first described in Ref. [30]. Although it has no analytical solution, this problem is useful in testing the performance of a method in handling unsteady shock interactions in multiple dimensions. The 2D rectangular domain is three units wide and one unit high. The step is between $x_1=0.6$ and $x_1=3.0$, with a height of 0.2 units. As boundary condition at the step, the normal component of the velocity is set to zero. The normal component of the velocity is zero also along the upper and lower channel walls. The supersonic inflow conditions at the left boundary are $\rho=1.4$, $p=1.0$, $u_1=3.0$, and $u_2=0.0$, corresponding to a Mach number of 3. Because the condition at the outflow boundary on the right is supersonic throughout the calculation, no boundary condition is specified there. The initial conditions are set equal to the inflow conditions, with $u_1=0.0$ along the left edge of the step. The mesh, which is shown in Fig. 3, has 58,509 nodes, 173,130 elements, and 290,142 edges.

The time-step size is $10^{-3}$, and the test duration is 2.0. For the preconditioned GMRES solver, 30 vectors are used in the Krylov basis, with a maximum of 100 cycles and a tolerance of $10^{-5}$.

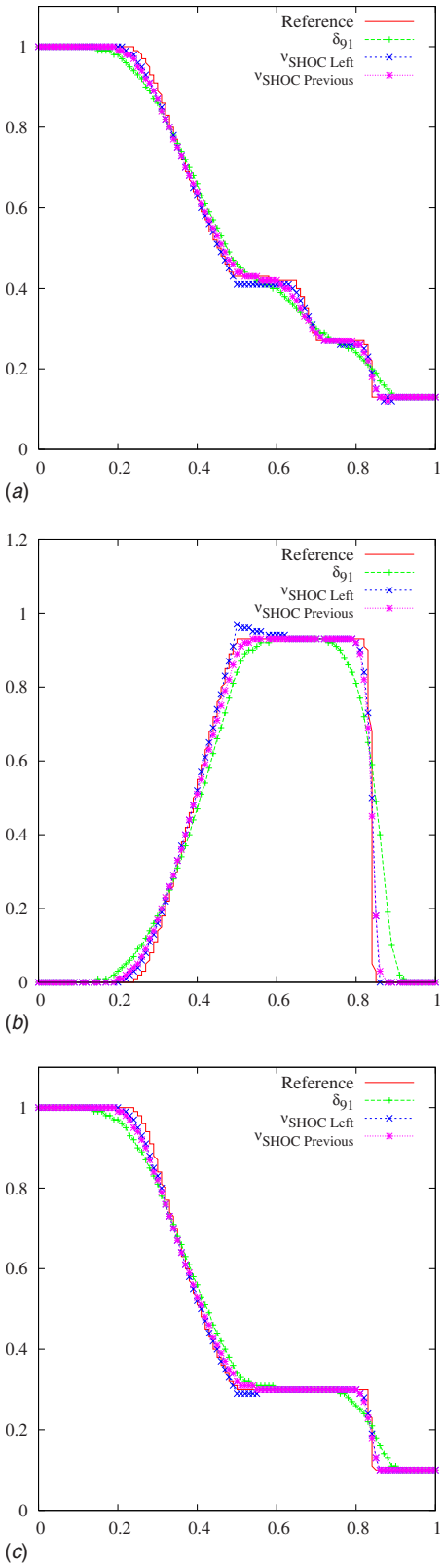



**Fig. 2 1D shock tube. Solutions at $t=0.2$. (*a*) Density, (*b*) velocity, and (*c*) pressure.**

Nonlinear tolerance is $10^{-2}$. As reference values in Eq. (16), we use the initial conditions. Figures 4 and 5 show the density obtained with $\delta_{91}$ and $\nu_{\text{SHOC}}$. We note that they show good agreement with the benchmark solution [30].

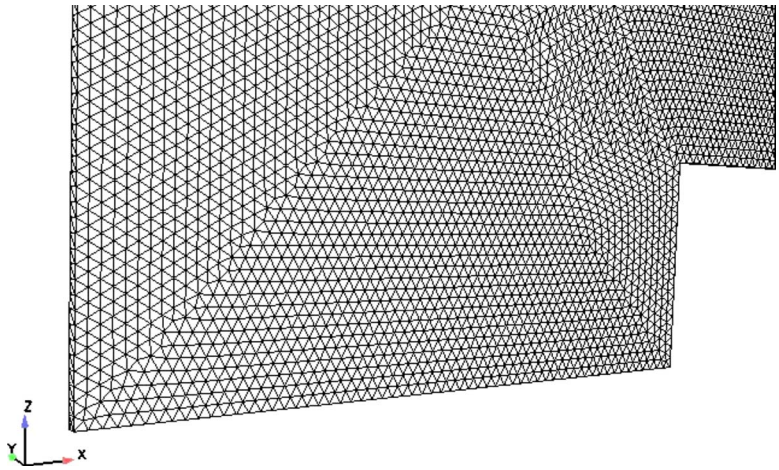



**Fig. 3 2D flow in a channel with a step. Mesh until $x_1=0.7$.**

Figures 6 and 7 show the distribution of $\delta_{91}$ and $\nu_{\mathrm{SHOC}}$. We note that the two parameters have the same order of magnitude ($0<\delta_{91}<2.75\times10^{-2}$ and $0<\nu_{\mathrm{SHOC}}<2.0\times10^{-2}$) but have different distributions, with $\nu_{\mathrm{SHOC}}$ mimicking the shock interactions.

**4.3 3D Flow Around a Sphere.** The sphere has a unit radius and the Mach number is 3. We consider only half of the sphere to obtain the steady-state solution. Figure 8 shows the dimensions of the problem domain.

The far-field conditions are $\rho_\infty=1$, $\mathbf{u}_\infty=(3,0,0)^T$, and $e_\infty=6.3$. We impose these far-field conditions at the left and upper boundaries. The flow is supersonic at the outflow boundary on the right, and therefore no boundary condition is specified there. Along the symmetry plane and on the cylinder surface the normal component of the velocity is zero. The initial conditions are set to the far-field conditions. The mesh, shown in Fig. 9, has 15,032 nodes, 78,915 elements, and 97,809 edges.

The time-step size is $10^{-2}$, and the time-marching continues until $t=6$. For the preconditioned GMRES solver, 25 vectors are used in the Krylov basis, with a maximum of 50 cycles and a tolerance of $10^{-3}$. The number of nonlinear iterations per time step is 3. As reference values in Eq. (16), we use the initial conditions. Figures 10 and 11 show the density obtained with $\delta_{91}$ and $\nu_{\mathrm{SHOC}}$. The two are in good agreement.

Table 2 shows, for the element- and edge-based data structures, the relative CPU times for the matrix evaluation, matrix-vector

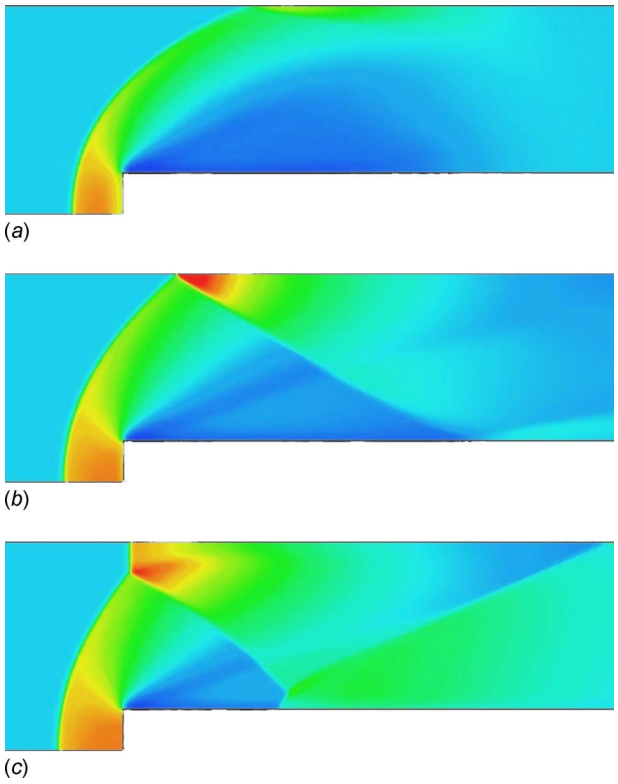



**Fig. 4 2D flow in a channel with a step. Density obtained with $\delta_{91}$. (*a*) $t=0.3$, (*b*) $t=0.6$, and (*c*) $t=1.2$.**

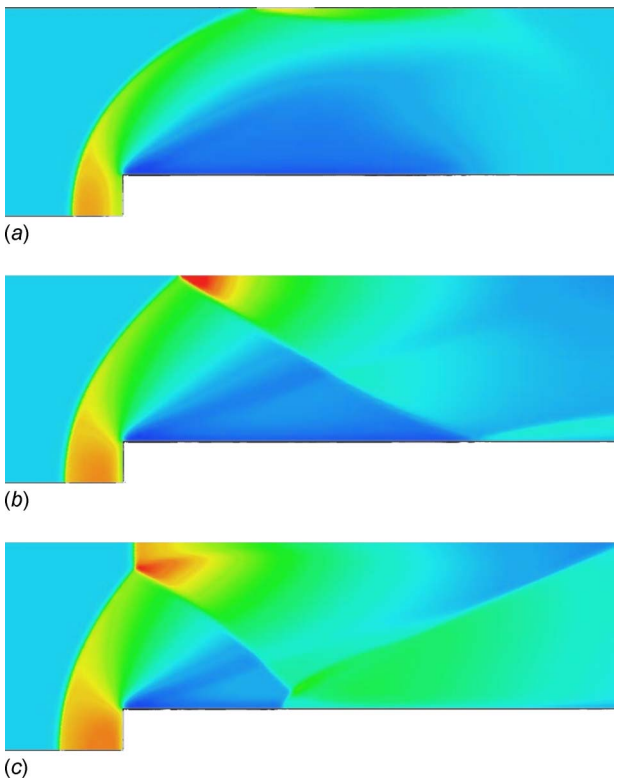



**Fig. 5 2D flow in a channel with a step. Density obtained with $\nu_{\mathrm{SHOC}}$. (*a*) $t=0.3$, (*b*) $t=0.6$, and (*c*) $t=1.2$.**

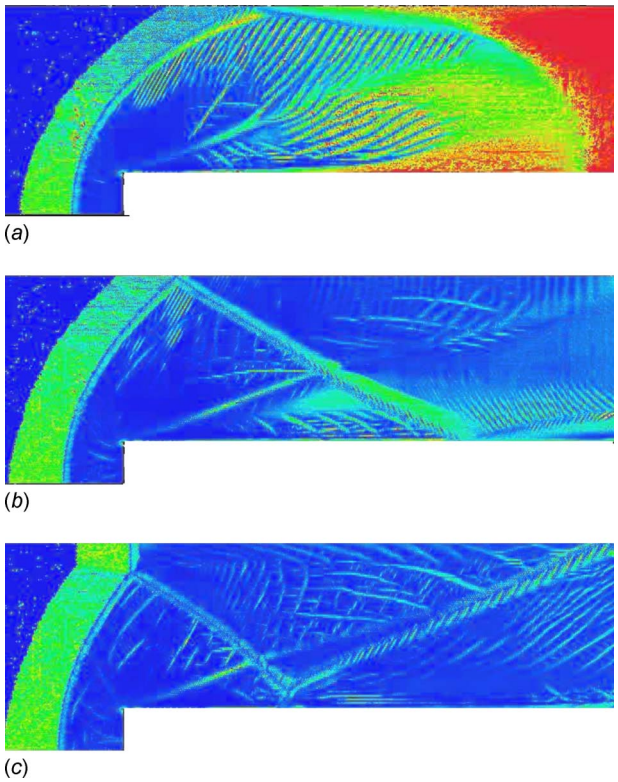


Fig. 6 2D flow in a channel with a step. Distribution of $\delta_{91}$. (*a*) $t=0.3$, (*b*) $t=0.6$, and (*c*) $t=1.2$.

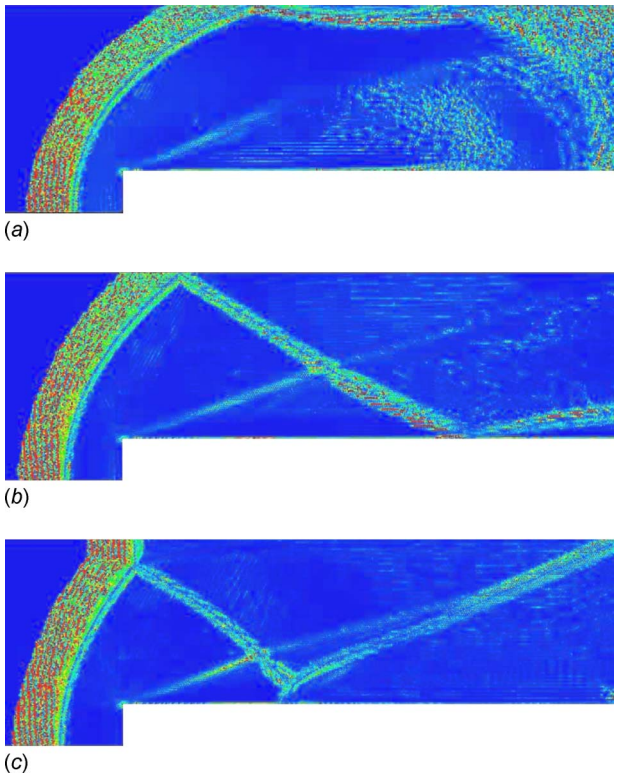


Fig. 7 2D flow in a channel with a step. Distribution of $\nu_{SHOC}$. (*a*) $t=0.3$, (*b*) $t=0.6$, and (*c*) $t=1.2$.

products in the GMRES iterations, and the total run times. Matrix evaluation is somewhat slower for the edge-based data structure, but this is very much compensated by the large reduction in the CPU time involved in the matrix-vector products, resulting in roughly 40% reduction in the total run time.

## 5 Concluding Remarks

We provided a comprehensive assessment of the stabilization and shock-capturing parameters introduced recently for SUPG formulation of compressible flows based on conservation variables. We focused on performance evaluation of the $YZ\beta$ shock-capturing parameter. We used well-known 1D, 2D, and 3D test problems with tetrahedral meshes. At each time step, the coupled nonlinear equations involved were solved with the predictor-multicorrector algorithm. An iterative technique with nodal-block-diagonal preconditioner and GMRES update method was employed for solving the linear equation system involved. We used an edge-based data structure to store the Jacobian and perform the matrix-vector products. In our test computations, we compared the $YZ\beta$ shock-capturing parameter to $\delta_{91}$, which was derived from its counterpart in entropy variables. We also tested different options for choosing the reference values used in $YZ\beta$ shock-capturing. In addition to being simpler and requiring less floating point operations, $YZ\beta$ shock-capturing yields better shock quality. We provided an assessment of the computational efficiency of the edge-based structure compared with the element-based one. Although computing and storing the edge-matrices is a bit slower, matrix-vector products are computed around five times faster, reducing the total run time by about 40%.


## Acknowledgment

We are indebted to Jose J. Camata, a graduate student at COPPE/UFRJ, for his help in the last stages of this work.

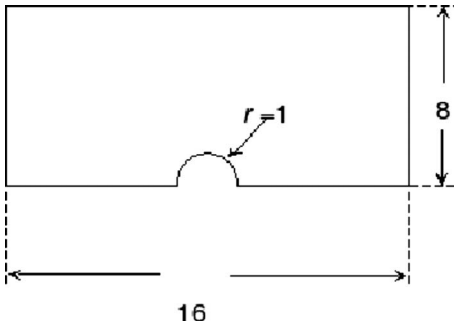




Fig. 8 3D flow around a sphere. Dimensions of the problem domain. The cylinder is located at $x_1=8$.

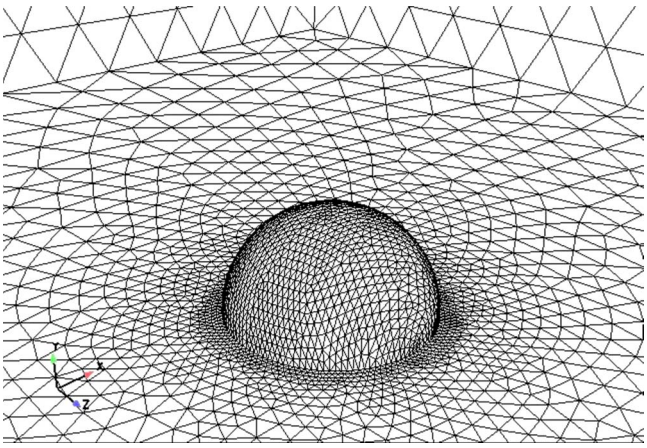


Fig. 9 3D flow around a sphere. Mesh.

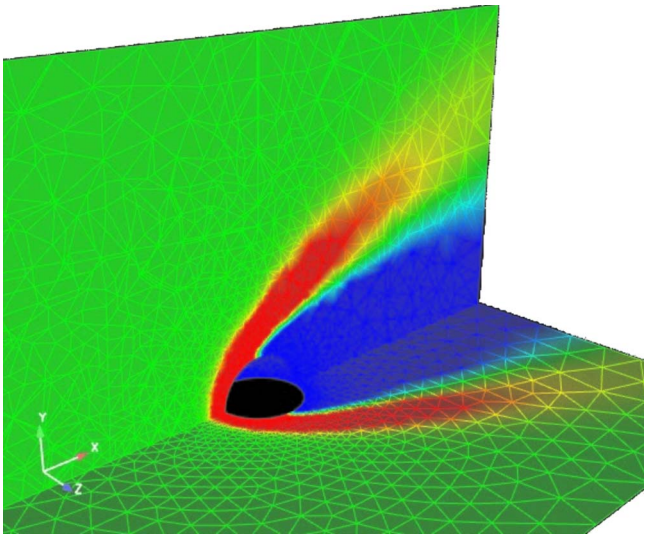


**Fig. 10 3D flow around a sphere. Density obtained with $\delta_{91}$.**

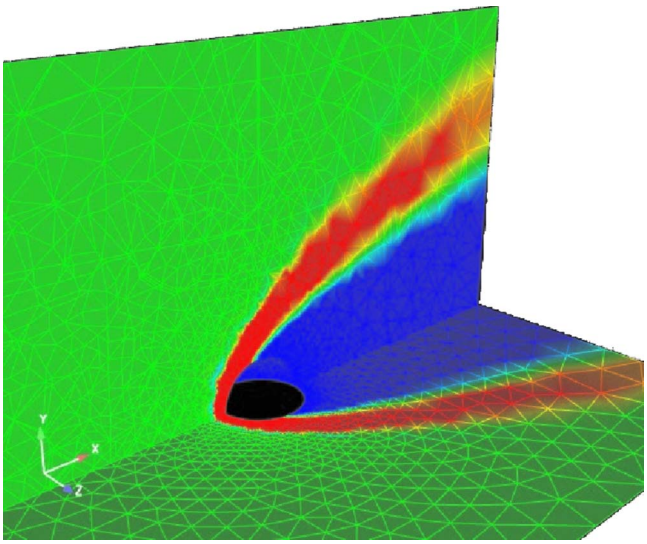


**Fig. 11 3D flow around a sphere. Density obtained with $\nu_{SHOC}$.**

**Table 2 3D flow around a sphere. Relative CPU time for the element- and edge-based data structures.**

| | Matrix evaluation | Matrix-vector | Total run |
|---|---|---|---|
| Element | 0.831 | 1.000 | 1.000 |
| Edge | 1.000 | 0.198 | 0.599 |

## References


[1] Hughes, T. J. R., and Brooks, A. N., 1979, "A Multi-Dimensional Upwind Scheme With No Crosswind Diffusion," in *Finite Element Methods for Convection Dominated Flows*, AMD-Vol. 34, T. J. R. Hughes, ed., ASME, New York, pp. 19–35.

[2] Brooks, A. N., and Hughes, T. J. R., 1982, "Streamline Upwind/Petrov–Galerkin Formulations for Convection Dominated Flows With Particular Emphasis on the Incompressible Navier–Stokes Equations," Comput. Methods Appl. Mech. Eng., **32**, pp. 199–259.

[3] Tezduyar, T. E., and Hughes, T. J. R., 1982, "Development of Time-Accurate Finite Element Techniques for First-Order Hyperbolic Systems With Particular Emphasis on the Compressible Euler Equations," NASA Technical Report No. NASA-CR-204772.

[4] Tezduyar, T. E., and Hughes, T. J. R., 1983, "Finite Element Formulations for Convection Dominated Flows With Particular Emphasis on the Compressible Euler Equations," AIAA Paper No. 83-0125.

[5] Hughes, T. J. R., and Tezduyar, T. E., 1984, "Finite Element Methods for First-Order Hyperbolic Systems With Particular Emphasis on the Compressible Euler Equations," Comput. Methods Appl. Mech. Eng., **45**, pp. 217–284.

[6] Hughes, T. J. R., Franca, L. P., and Mallet, M., 1987, "A New Finite Element Formulation for Computational Fluid Dynamics: VI. Convergence Analysis of the Generalized Supg Formulation for Linear Time-Dependent Multi-Dimensional Advective-Diffusive Systems," Comput. Methods Appl. Mech. Eng., **63**, pp. 97–112.

[7] Le Beau, G. J., and Tezduyar, T. E., 1991, "Finite Element Computation of Compressible Flows With the SUPG Formulation," *Advances in Finite Element Analysis in Fluid Dynamics*, FED-Vol. 123, ASME, New York, pp. 21–27.

[8] Le Beau, G. J., Ray, S. E., Aliabadi, S. K., and Tezduyar, T. E., 1993, "SUPG Finite Element Computation of Compressible Flows With the Entropy and Conservation Variables Formulations," Comput. Methods Appl. Mech. Eng., **104**, pp. 397–422.

[9] Tezduyar, T. E., and Park, Y. J., 1986, "Discontinuity Capturing Finite Element Formulations for Nonlinear Convection-Diffusion-Reaction Equations," Comput. Methods Appl. Mech. Eng., **59**, pp. 307–325.

[10] Tezduyar, T. E., and Osawa, Y., 2000, "Finite Element Stabilization Parameters Computed From Element Matrices and Vectors," Comput. Methods Appl. Mech. Eng., **190**, pp. 411–430.

[11] Catabriga, L., Coutinho, A. L. G. A., and Tezduyar, T. E., 2005, "Compressible Flow SUPG Parameters Computed From Element Matrices," Commun. Numer. Methods Eng., **21**, pp. 465–476.

[12] Catabriga, L., Coutinho, A. L. G. A., and Tezduyar, T. E., 2006, "Compressible Flow SUPG Parameters Computed From Degree-of-Freedom Submatrices," Comput. Mech., **38**, pp. 334–343.

[13] Catabriga, L., Coutinho, A. L. G. A., and Tezduyar, T. E., 2004, "Compressible Flow SUPG Stabilization Parameters Computed From Element-Edge Matrices," Comput. Fluid Dyn. J., **13**, pp. 450–459.

[14] Catabriga, L., and Coutinho, A. L. G. A., 2002, "Implicit SUPG Solution of Euler Equations Using Edge-Based Data Structures," Comput. Methods Appl. Mech. Eng., **191**, pp. 3477–3490.

[15] Tezduyar, T. E., 2004, "Finite Element Methods for Fluid Dynamics With Moving Boundaries and Interfaces," *Encyclopedia of Computational Mechanics*, Vol. 3 (Fluids), E. Stein, R. De Borst, and T. J. R. Hughes, eds., Wiley, New York, Chap. 17.

[16] Tezduyar, T. E., 2004 "Determination of the Stabilization and Shock-Capturing Parameters in SUPG Formulation of Compressible Flows," *Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2004*, Jyvaskyla, Finland.

[17] Tezduyar, T. E., 2007, "Finite Elements in Fluids: Stabilized Formulations and Moving Boundaries and Interfaces," Comput. Fluids, **36**, pp. 191–206.

[18] Tezduyar, T. E., and Senga, M., 2006, "Stabilization and Shock-Capturing Parameters in SUPG Formulation of Compressible Flows," Comput. Methods Appl. Mech. Eng., **195**, pp. 1621–1632.

[19] Tezduyar, T. E., and Senga, M., 2007, "SUPG Finite Element Computation of Inviscid Supersonic Flows With $YZ\beta$ Shock-Capturing," Comput. Fluids, **36**, pp. 147–159.

[20] Tezduyar, T. E., Senga, M., and Vicker, D., 2006, "Computation of Inviscid Supersonic Flows Around Cylinders and Spheres With the Supg Formulation and $YZ\beta$ Shock-Capturing," Comput. Mech., **38**, pp. 469–481.

[21] Rispoli, F., Saavedra, R., Corsini, A., and Tezduyar, T. E., 2007, "Computation of Inviscid Compressible Flows With the V-SGS Stabilization and $YZ\beta$ Shock-Capturing," Int. J. Numer. Methods Fluids, **54**, pp. 695–706.

[22] Corsini, A., Rispoli, F., and Santoriello, A., 2005, "A Variational Multiscale High-Order Finite Element Formulation for Turbomachinery Flow Computations," Comput. Methods Appl. Mech. Eng., **194**, pp. 4797–4823.

[23] Rispoli, F., and Saavedra, R., 2006, "A Stabilized Finite Element Method Based on SGS Models for Compressible Flows," Comput. Methods Appl. Mech. Eng., **196**, pp. 652–664.

[24] Saad, Y., and Schultz, M., 1986, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," SIAM (Soc. Ind. Appl. Math.) J. Sci. Stat. Comput., **7**, pp. 856–869.

[25] Aliabadi, S. K., Ray, S. E., and Tezduyar, T. E., 1993, "SUPG Finite Element Computation of Compressible Flows With the Entropy and Conservation Variables Formulations," Comput. Mech., **11**, pp. 300–312.

[26] Souza, D. A. F., Martins, M. A. D., and Coutinho, A. L. G. A., 2005, "Edge-Based Adaptive Implicit/Explicit Finite Element Procedures for Three-Dimensional Transport Problems," Commun. Numer. Methods Eng., **21**, pp. 545–552.

[27] Lohner, R., 2001, *Applied CFD Techniques: An Introduction Based on Finite Element Methods*, Springer-Verlag, Berlin /Wiley, New York.

[28] Coutinho, A. L. G. A., Martins, M. A. D., Sydenstricker, R. M., and Elias, R. N., 2006, "Performance Comparison of Data-Reordering Algorithms for Sparse Matrix-Vector Multiplication in Edge-Based Unstructured Grid Computations," Int. J. Numer. Methods Eng., **66**, pp. 431–460.

[29] Luo, H., Baum, J. D., and Lohner, R., 2006, "A Hybrid Cartesian Grid and Gridless Method for Compressible Flow," J. Comput. Phys., **214**, pp. 618–632.

[30] Emery, A. F., 1968, "An Evaluation of Several Differencing Methods for Inviscid Fluid Flow Problems," J. Comput. Phys., **2**, pp. 306–331.

**Franco Rispoli**
Dipartimento di Meccanica e Aeronautica,
Università degli Studi di Roma "La Sapienza,"
Via Eudossiana 18,
I-00184 Roma, Italy
e-mail: franco.rispoli@uniroma1.it

**Rafael Saavedra**
Departamento de Ingeneria Mecánica-Electrica,
Universidad de Piura,
Avenida Ramón Mugica 131,
Piura, Perú
e-mail: rsaavedr@udep.edu.pe

**Filippo Menichini**
Dipartimento di Meccanica e Aeronautica,
Università degli Studi di Roma "La Sapienza,"
Via Eudossiana 18,
I-00184 Roma, Italy
e-mail: f.menichini@dma.ing.uniroma1.it

**Tayfun E. Tezduyar**
Mechanical Engineering,
Rice University,
MS 321,
6100 Main Street,
Houston, TX 77005
e-mail: tezduyar@rice.edu

# Computation of Inviscid Supersonic Flows Around Cylinders and Spheres With the V-SGS Stabilization and *YZβ* Shock-Capturing

*The YZβ shock-capturing technique was introduced originally for use in combination with the streamline-upwind/Petrov–Galerkin (SUPG) formulation of compressible flows in conservation variables. It is a simple residual-based shock-capturing technique. Later it was also combined with the variable subgrid scale (V-SGS) formulation of compressible flows in conservation variables and tested on standard 2D test problems. The V-SGS method is based on an approximation of the class of SGS models derived from the Hughes variational multiscale method. In this paper, we carry out numerical experiments with inviscid supersonic flows around cylinders and spheres to evaluate the performance of the YZβ shock-capturing combined with the V-SGS method. The cylinder computations are carried out at Mach numbers 3 and 8, and the sphere computations are carried out at Mach number 3. The results compare well to those obtained with the YZβ shock-capturing combined with the SUPG formulation, which were shown earlier to compare very favorably to those obtained with the well established* OVERFLOW *code.*
[DOI: 10.1115/1.3057496]

*Keywords: supersonic flows, variable subgrid scale formulation, SUPG formulation, YZβ shock-capturing, cylinders and spheres*

## 1 Introduction

The streamline-upwind/Petrov–Galerkin (SUPG) formulation of compressible flows was first introduced in 1982, soon after the introduction of the SUPG formulation of incompressible flows [1,2]. This first SUPG formulation of compressible flows was in the context of conservation variables (see Refs. [3,4]). It did not involve any shock-capturing term. The test computations clearly showed the need for extra measures at the shocks. The formulation was later recast in entropy variables and supplemented with a shock-capturing term [5]. This resulted in better shock profiles. In a 1991 ASME paper [6], the SUPG formulation introduced in Refs. [3,4] was supplemented with a very similar shock-capturing term, which included a shock-capturing parameter that is now called "$\delta_{91}$." This shock-capturing parameter was derived from the one given in Ref. [5] for the entropy variables. It was shown in Ref. [6] that with the added shock-capturing term, the original SUPG formulation of compressible flows in conservation variables is very comparable in accuracy to the SUPG formulation in entropy variables. Shortly after that, the 2D test computations for inviscid flows reported in Ref. [7] showed that the SUPG formulation in conservation and entropy variables yielded indistinguishable results.

The *YZβ* shock-capturing, which was introduced in Refs. [8,9], is based on using a simple residual-based shock-capturing parameter. The new parameter is less costly to compute with than $\delta_{91}$. It has options for smoother or sharper computed shocks. A preliminary set of test computations with the *YZβ* shock-capturing was reported in Ref. [10] for inviscid supersonic flows. These were standard 2D test problems with very simple geometries, and the meshes were made of quadrilateral elements. For the same 2D test problems, a more comprehensive set of computations with different element types and mesh orientations was reported in Ref. [11]. In Ref. [12], numerical experiments were carried out for inviscid supersonic flows around cylinders and spheres. The objective was to evaluate the performance of the *YZβ* shock-capturing in problems that are more challenging because of blunt geometries and high Mach numbers. The *YZβ* results were compared with those obtained with $\delta_{91}$. For 2D structured meshes, the *YZβ* results were also compared with those obtained with the OVERFLOW code [13]. All those test computations showed that the *YZβ* shock-capturing parameters are not only much simpler than $\delta_{91}$ but also superior in accuracy.

The variable subgrid scale (V-SGS) method was first introduced in Ref. [14] for the advection-diffusion-reaction equation and for incompressible flows. It is based on an approximation of the class of SGS models derived from the Hughes variational multiscale (Hughes-VMS) method [15]. In Ref. [16], the V-SGS approach was formulated for compressible flows in conservation variables. In Ref. [17], the *YZβ* shock-capturing was used in combination the V-SGS method. The test problems computed in Ref. [17] were the same as the standard 2D test problems mentioned in the previous paragraph. The results reported in Ref. [17] show that the *YZβ* shock-capturing yields better performance also when it is used in conjunction with the V-SGS method. In this paper, we evaluate the performance of the *YZβ* shock-capturing combined with the V-SGS method by carrying out numerical experiments with inviscid supersonic flows around cylinders and spheres. The test problems are basically the cylinder and sphere test problems mentioned in the previous paragraph. In Sec. 2 we review the governing equations of compressible flows in conservation variables. The SUPG and V-SGS formulations are described in Sec. 3,

Contributed by the Applied Mechanics Division of ASME for publication in the JOURNAL OF APPLIED MECHANICS. Manuscript received November 21, 2007; final manuscript received July 9, 2008; published online January 26, 2009. Review conducted by Arif Masud.

and the $YZ\beta$ shock-capturing is discussed in Sec. 4. The test computations are presented in Sec. 5, and the concluding remarks are given in Sec. 6.

## 2 Navier–Stokes Equations of Compressible Flows

Let $\Omega \subset \mathbb{R}^{n_{sd}}$ be the spatial domain with boundary $\Gamma$, and let $(0,T)$ be the time domain. The symbols $\rho$, $\mathbf{u}$, $p$, and $e$ will represent the density, velocity, pressure, and total energy, respectively. The Navier–Stokes equations of compressible flows can be written on $\Omega$ and $\forall t \in (0,T)$ as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} - \frac{\partial \mathbf{E}_i}{\partial x_i} - \mathbf{R} = \mathbf{0} \tag{1}$$

where $\mathbf{U} = (\rho, \rho u_1, \rho u_2, \rho u_3, \rho e)^T$ is the vector of conservation variables and $\mathbf{F}_i$ and $\mathbf{E}_i$ are, respectively, the Euler and viscous flux vectors,

$$\mathbf{F}_i = \begin{pmatrix} u_i \rho \\ u_i \rho u_1 + \delta_{i1} p \\ u_i \rho u_2 + \delta_{i2} p \\ u_i \rho u_3 + \delta_{i3} p \\ u_i (\rho e + p) \end{pmatrix}, \quad \mathbf{E}_i = \begin{pmatrix} 0 \\ T_{i1} \\ T_{i2} \\ T_{i3} \\ -q_i + T_{ik} u_k \end{pmatrix} \tag{2}$$

Here $\delta_{ij}$ are the components of the identity tensor $\mathbf{I}$, $q_i$ are the components of the heat flux vector, and $T_{ij}$ are the components of the Newtonian viscous stress tensor,

$$\mathbf{T} = \lambda (\nabla \cdot \mathbf{u}) \mathbf{I} + 2\mu \boldsymbol{\varepsilon}(\mathbf{u}) \tag{3}$$

where $\lambda$ and $\mu (= \rho \nu)$ are the viscosity coefficients, $\nu$ is the kinematic viscosity, and $\boldsymbol{\varepsilon}(\mathbf{u})$ is the strain-rate tensor,

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \tfrac{1}{2}((\nabla \mathbf{u}) + (\nabla \mathbf{u})^T) \tag{4}$$

It is assumed that $\lambda = -2\mu/3$. The equation of state used here corresponds to the ideal gas assumption. The term $\mathbf{R}$ represents all other components that might enter the equations, including the external forces.

Equation (1) can further be written in the following form:

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}_i \frac{\partial \mathbf{U}}{\partial x_i} - \frac{\partial}{\partial x_i}\left( \mathbf{K}_{ij} \frac{\partial \mathbf{U}}{\partial x_j} \right) - \mathbf{R} = \mathbf{0} \tag{5}$$

where

$$\mathbf{A}_i = \frac{\partial \mathbf{F}_i}{\partial \mathbf{U}}, \quad \mathbf{K}_{ij} \frac{\partial \mathbf{U}}{\partial x_j} = \mathbf{E}_i \tag{6}$$

Appropriate sets of boundary and initial conditions are assumed to accompany Eq. (5).

## 3 SUPG and V-SGS Stabilizations

In describing the SUPG and V-SGS formulations of Eq. (5), we assume that we have constructed some suitably defined finite-dimensional trial solution and test function spaces $\mathcal{S}_{\mathbf{U}}^h$ and $\mathcal{V}_{\mathbf{U}}^h$. Based on that, the SUPG [3,4,6] and V-SGS [16] formulations can be written as follows: find $\mathbf{U}^h \in \mathcal{S}_{\mathbf{U}}^h$ such that $\forall \mathbf{W}^h \in \mathcal{V}_{\mathbf{U}}^h$,

$$\begin{aligned} &\int_\Omega \mathbf{W}^h \cdot \left( \frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i} \right) d\Omega \\ &\quad + \int_\Omega \left( \frac{\partial \mathbf{W}^h}{\partial x_i} \right) \cdot \left( \mathbf{K}_{ij}^h \frac{\partial \mathbf{U}^h}{\partial x_j} \right) d\Omega - \int_{\Gamma_H} \mathbf{W}^h \cdot \mathbf{H}^h d\Gamma \\ &\quad - \int_\Omega \mathbf{W}^h \cdot \mathbf{R}^h d\Omega + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \mathbf{P}_{\text{stab}}(\mathbf{W}^h) \\ &\quad \cdot \left[ \frac{\partial \mathbf{U}^h}{\partial t} + \mathbf{A}_i^h \frac{\partial \mathbf{U}^h}{\partial x_i} - \frac{\partial}{\partial x_i}\left( \mathbf{K}_{ij}^h \frac{\partial \mathbf{U}^h}{\partial x_j} \right) - \mathbf{R}^h \right] d\Omega \\ &\quad + \sum_{e=1}^{n_{el}} \int_{\Omega^e} \nu_{\text{shoc}} \left( \frac{\partial \mathbf{W}^h}{\partial x_i} \right) \cdot \left( \frac{\partial \mathbf{U}^h}{\partial x_i} \right) d\Omega = 0 \end{aligned} \tag{7}$$

where $\mathbf{H}^h$ represents the natural boundary conditions associated with Eq. (5) and $\Gamma_H$ is the part of the boundary where such boundary conditions are specified. The vector operator $\mathbf{P}_{\text{stab}}(\mathbf{W}^h)$ takes the following forms for the SUPG and V-SGS stabilizations, respectively:

$$\mathbf{P}_{\text{stab}}(\mathbf{W}^h) = \mathbf{P}_{\text{SUPG}}(\mathbf{W}^h) \tag{8}$$

$$\mathbf{P}_{\text{stab}}(\mathbf{W}^h) = \mathbf{P}_{\text{VSGS}}(\mathbf{W}^h) \tag{9}$$

where

$$\mathbf{P}_{\text{SUPG}}(\mathbf{W}^h) = \left[ \boldsymbol{\tau}_{\text{SUPG}} \left( \frac{\partial \mathbf{W}^h}{\partial x_k} \right) \right] \mathbf{A}_k^h \tag{10}$$

$$\mathbf{P}_{\text{VSGS}}(\mathbf{W}^h) = \left[ (\mathbf{A}_k^h)^T \left( \frac{\partial \mathbf{W}^h}{\partial x_k} \right) + \frac{\partial}{\partial x_l} \left( (\mathbf{K}_{lk}^h)^T \left( \frac{\partial \mathbf{W}^h}{\partial x_k} \right) \right) \right] \boldsymbol{\tau}_{\text{VSGS}} \tag{11}$$

The diagonal matrices $\boldsymbol{\tau}_{\text{SUPG}}$ and $\boldsymbol{\tau}_{\text{VSGS}}$ are the SUPG and V-SGS stabilization parameters. The expression for $\boldsymbol{\tau}_{\text{SUPG}}$ can be found in Refs. [8–12].

The expression for $\boldsymbol{\tau}_{\text{VSGS}}$ used in the computations reported here is a modified version of the expression given in Ref. [16]. We describe those modifications here with the notation used in Ref. [16]. In calculating the components of $\boldsymbol{\tau}_{\text{VSGS}}$, namely, $\tau_{\text{VSGS}}^{\rho}$, $\tau_{\text{VSGS}}^{u}$, and $\tau_{\text{VSGS}}^{e}$, instead of using the expressions given by Eqs. (62) and (63) in Ref. [16], we use the "$r$-switch" [18] combination of their directional components,

$$\tau_{\text{VSGS}}^m = \left( \frac{1}{(\tau_{\xi_1}^m)^r} + \frac{1}{(\tau_{\xi_2}^m)^r} + \frac{1}{(\tau_{\xi_3}^m)^r} \right)^{-1/r}, \quad \forall\, m \in \{\rho, u, e\} \tag{12}$$

where $\xi_i$s are the element coordinates and, typically, $r=2$. The directional stabilization parameters for the momentum and energy balance equations are given as

$$\tau_{\xi_i}^m = (\tau_{\text{SC}}^m)_{\xi_i} (1 + f(\xi_i, \text{Pe}_{\xi_i}^m)), \quad \forall\, m \in \{u, e\} \tag{13}$$

where $(\tau_{\text{SC}}^m)_{\xi_i}$ is the mean value of the directional stabilization parameter, $\text{Pe}_{\xi_i}^m$s are the directional element Peclet numbers, and the functional form $f(\xi_i, \text{Pe}_{\xi_i}^m)$ is given in Ref. [16]. The directional stabilization parameter for the mass balance equation is given as

$$\tau_{\xi_i}^{\rho} = (\tau_{\text{SC}}^{\rho})_{\xi_i} (1 + f(\xi_i)) \tag{14}$$

For inviscid flows, the directional stabilization parameters for the mass, momentum, and energy balance equations are all given as

$$\tau_{\xi_i}^m = (\tau_{\text{SC}}^m)_{\xi_i} (1 + f(\xi_i)), \quad \forall\, m \in \{\rho, u, e\} \tag{15}$$

For the functional form $f(\xi_i)$, instead of using the expression given by Eq. (78) in Ref. [16], we use the following expression:

$$f(\xi_i) = -\frac{u_{\xi_i}}{|u_{\xi_i}|} \xi_i \quad \text{(no sum)} \tag{16}$$

where $u_{\xi_i}$s are the advection components defined along the element coordinates. We also modify how those advection components are defined. Instead of using the expression given by Eq. (71) in Ref. [16], we calculate them by solving the following equation system:

$$(\mathbf{e}_{\xi_1} \cdot \mathbf{e}_{\xi_1}) u_{\xi_1} + (\mathbf{e}_{\xi_1} \cdot \mathbf{e}_{\xi_2}) u_{\xi_2} + (\mathbf{e}_{\xi_1} \cdot \mathbf{e}_{\xi_3}) u_{\xi_3} = \mathbf{e}_{\xi_1} \cdot \boldsymbol{\lambda}$$

$$(\mathbf{e}_{\xi_2} \cdot \mathbf{e}_{\xi_1}) u_{\xi_1} + (\mathbf{e}_{\xi_2} \cdot \mathbf{e}_{\xi_2}) u_{\xi_2} + (\mathbf{e}_{\xi_2} \cdot \mathbf{e}_{\xi_3}) u_{\xi_3} = \mathbf{e}_{\xi_2} \cdot \boldsymbol{\lambda}$$

$$(\mathbf{e}_{\xi_3}\cdot\mathbf{e}_{\xi_1})u_{\xi_1}+(\mathbf{e}_{\xi_3}\cdot\mathbf{e}_{\xi_2})u_{\xi_2}+(\mathbf{e}_{\xi_3}\cdot\mathbf{e}_{\xi_3})u_{\xi_3}=\mathbf{e}_{\xi_3}\cdot\boldsymbol{\lambda} \tag{17}$$

where $\mathbf{e}_{\xi_i}$s are unit vectors along the element coordinates and $\boldsymbol{\lambda}$ is a vector constructed in Ref. [16] from the eigenvalue definitions given in Ref. [16] for the Euler (inviscid) part of the system given by Eq. (5). For completeness, we provide here the definition of $\boldsymbol{\lambda}$,

$$\boldsymbol{\lambda}=\left(1+\frac{1}{M}\right)\mathbf{u} \tag{18}$$

where $M$ is the Mach number. The shock-capturing parameter is denoted by $\nu_{\text{shoc}}$. It was discussed briefly in Sec. 1 and will further be discussed in Sec. 4.

## 4 $YZ\beta$ Shock-Capturing

In the "YZ" version of the $YZ\beta$ shock-capturing, $\nu_{\text{shoc}}$ was defined in Refs. [8–12] as

$$\nu_{\text{shoc}}=\|\mathbf{Y}^{-1}\mathbf{Z}\|\left(\sum_{i=1}^{n_{\text{sd}}}\left\|\mathbf{Y}^{-1}\frac{\partial\mathbf{U}^h}{\partial x_i}\right\|^2\right)^{\beta/2-1}\left(\frac{h_{\text{shoc}}}{2}\right)^{\beta} \tag{19}$$

This is the version we use in the computations we report in this paper. In Eq. (19), $\mathbf{Y}$ is a diagonal scaling matrix constructed from the reference values of the components of $\mathbf{U}$,

$$\mathbf{Y}=\begin{bmatrix}(U_1)_{\text{ref}} & 0 & 0 & 0 & 0\\ 0 & (U_2)_{\text{ref}} & 0 & 0 & 0\\ 0 & 0 & (U_3)_{\text{ref}} & 0 & 0\\ 0 & 0 & 0 & (U_4)_{\text{ref}} & 0\\ 0 & 0 & 0 & 0 & (U_5)_{\text{ref}}\end{bmatrix} \tag{20}$$

$$\mathbf{Z}=\mathbf{A}_i^h\frac{\partial\mathbf{U}^h}{\partial x_i} \tag{21}$$

and

$$h_{\text{shoc}}=h_{\text{JGN}}=2\left(\sum_{a=1}^{n_{en}}|\mathbf{j}\cdot\boldsymbol{\nabla}N_a|\right)^{-1} \tag{22}$$

where

$$\mathbf{j}=\frac{\boldsymbol{\nabla}\rho^h}{\|\boldsymbol{\nabla}\rho^h\|} \tag{23}$$

The parameter $\beta$ is set as $\beta=1$ for smoother computed shocks and as $\beta=2$ for sharper shocks.

*Remark 1.* When the expression given by Eq. (19) was originally introduced in Refs. [8,9], the intent was to have $\mathbf{Z}$ represent the residual and thus make $\nu_{\text{shoc}}$ residual based. This point was made explicitly in Refs. [11,12] by stating that the $YZ\beta$ shock-capturing parameters were "based on scaled residuals." This was the motivation behind the term $\|\mathbf{Y}^{-1}\mathbf{Z}\|$ in the expression. The selections given in Refs. [8,9] for $\mathbf{Z}$ represent the steady-state and time-dependent versions of the residual for inviscid flows with no source or external-force terms. The terms with the exponents $\beta/2-1$ and $\beta$ generate the correct local length scale.

Versions of $\nu_{\text{shoc}}$ that take into account the Mach number and shock intensity across a shock were proposed in Refs. [11,12]. In these references, $\nu_{\text{shoc}}$ given by Eq. (19) is modified as follows:

$$\nu_{\text{shoc}}\leftarrow\nu_{\text{shoc}}\left(1+\left(\frac{\|\boldsymbol{\nabla}\rho^h\|h_{\text{shoc}}}{\rho_{\text{ref}}}\right)\langle M^{1/b_{\text{M}}}-1\rangle\right)^{2/b_F} \tag{24}$$

where "$\langle\cdots\rangle$" is the Macaulay bracket,

$$\langle x-y\rangle=\begin{cases}0, & x\le y\\ x-y, & x>y\end{cases} \tag{25}$$

The reference density $\rho_{\text{ref}}$ is defined as

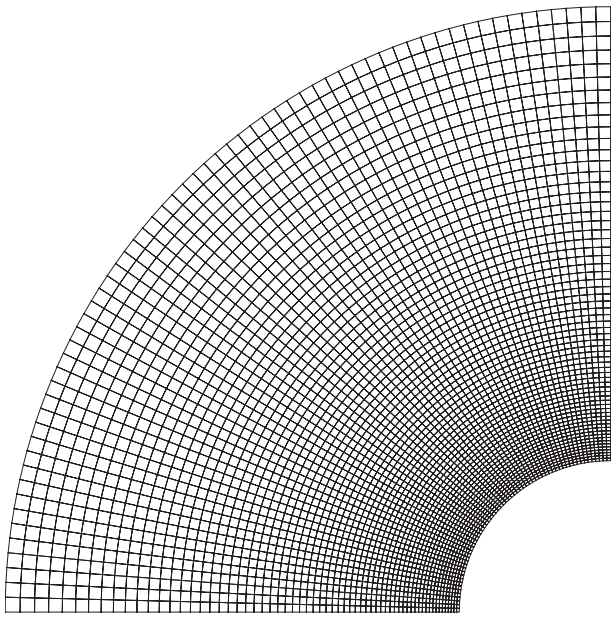


**Fig. 1 2D flow around a cylinder. Structured mesh with 4096 quadrilateral elements and 4225 nodes.**

$$\rho_{\text{ref}}=\rho_{\text{inf}}\left(\frac{\rho_{\text{sca}}}{\rho_{\text{inf}}}\right)^{b_R/2} \tag{26}$$

where $\rho_{\text{inf}}$ is the density at the inflow and $\rho_{\text{sca}}$ is a scaling density. In defining $\rho_{\text{sca}}$, one of the options considered in Refs. [8–12] was $\rho_{\text{sca}}=\rho_{\text{inf}}$. For flows with supersonic inflow and shocks, the other options considered were $\rho_{\text{sca}}=\rho_2$ (see Refs. [11,12]) and $\rho_{\text{sca}}=\rho_2-\rho_1$ (see Ref. [12]), where $\rho_1$ and $\rho_2$ are the density values before and after a normal shock corresponding to the inflow Mach number. The parameters $b_M$, $b_F$, and $b_R$ can each be set to 1 for smoother shocks and 2 for sharper shocks. Equation (24), without the exponent $2/b_F$, was originally introduced in Ref. [11]. With this expression, the definition of the shock-capturing viscosity takes into account the Mach number and shock intensity across a shock. The shock intensity is represented by the term $(\|\boldsymbol{\nabla}\rho^h\|h_{\text{shoc}}/\rho_{\text{ref}})$, which is a scaled measure of the jump in density. The Mach number is represented by the term $\langle M^{1/b_M}-1\rangle$, which becomes active for $M>1$.

## 5 Test Computations

As test problems, we consider steady-state cases in 2D and 3D. In all test computations, we use $\beta=1$. In Eq. (20), we set $(U_1)_{\text{ref}}$ to the inflow value of $\rho$, set $(U_2)_{\text{ref}}$, $(U_3)_{\text{ref}}$, and $(U_4)_{\text{ref}}$ to the inflow value of $\rho\|\mathbf{u}\|$, and set $(U_5)_{\text{ref}}$ to the inflow value of $\rho e$. In Eq. (24), unless stated otherwise, we set $b_F=2$.

**5.1 2D Flow Around a Cylinder.** We compute this case at $M=3$ and $M=8$, with a structured mesh. Figure 1 shows the mesh, which has 4096 quadrilateral elements and 4225 nodes. The test problem and the mesh are essentially identical to those reported in Ref. [12]. In computational units, the inflow density and velocity are 1.0 and (1,0). The inflow value of the total energy is 0.6984 at $M=3$ and 0.5279 at $M=8$. At the inflow boundary, we specify all conservation variables to be equal to their inflow values. On the cylinder surface and horizontal boundary, we specify the normal component of the velocity to be zero. No boundary conditions are specified at the outflow boundary.

Figures 2–4 show the results obtained for $M=3$. The results obtained with $b_M=2$ and $b_R=0$ are very close to those obtained with $b_M=1$, $b_R=1$, and $\rho_{\text{sca}}=\rho_2$. The results obtained with $b_M=1$

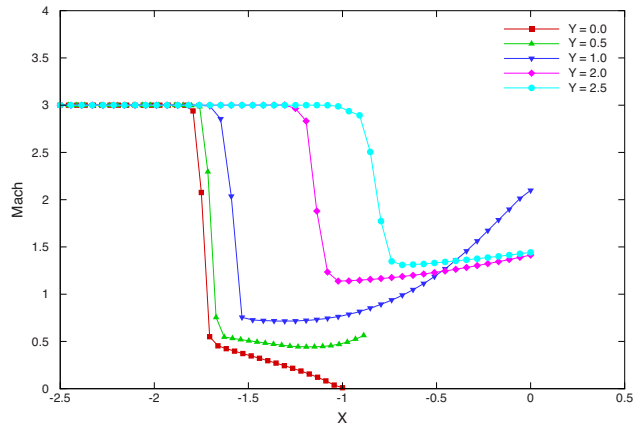



**Fig. 2 2D flow around a cylinder at $M=3$ Mach number. Computed with $b_M=2$ and $b_R=0$.**

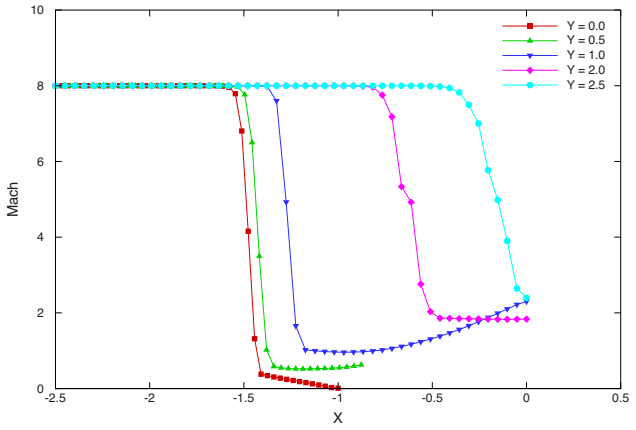



**Fig. 5 2D flow around a cylinder at $M=8$ Mach number. Computed with $b_M=2$ and $b_R=0$.**

and $b_R=0$ show slightly more dissipation than the other two $YZ\beta$ results. These results and trends are very close to those reported in Ref. [12].

Figures 5–7 show the results obtained for $M=8$. These results were obtained with the same three types of $YZ\beta$ shock-capturing we had for $M=3$. These results are also very close to those reported in Ref. [12].

**5.2 3D Flow Around a Sphere.** In this test problem, which is essentially the same as the one reported in Ref. [12], we have $M=3$. While an unstructured mesh was used in Ref. [12], here we use a block-structured mesh. Figure 8 shows the mesh, which has 52,896 hexahedral elements and 56,760 nodes. In computational units, the inflow density and velocity are 1.0 and $(0,-1,0)$. The inflow value of the total energy is 0.6984. At the inflow boundary, we specify all conservation variables to be equal to their inflow

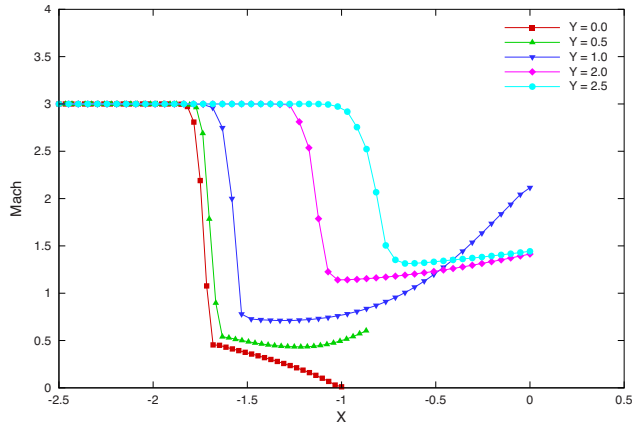



**Fig. 3 2D flow around a cylinder at $M=3$ Mach number. Computed with $b_M=1$ and $b_R=0$.**

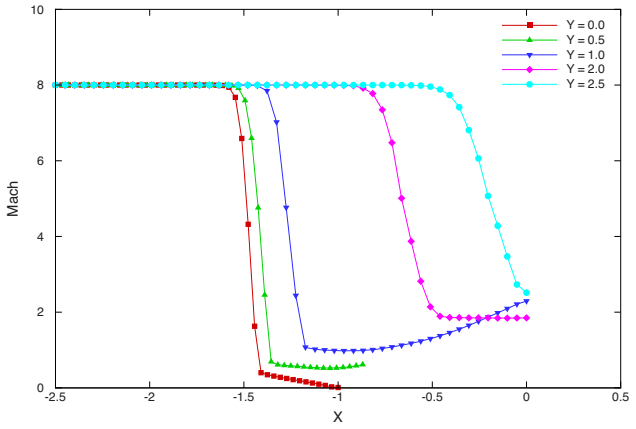



**Fig. 6 2D flow around a cylinder at $M=8$ Mach number. Computed with $b_M=1$ and $b_R=0$.**

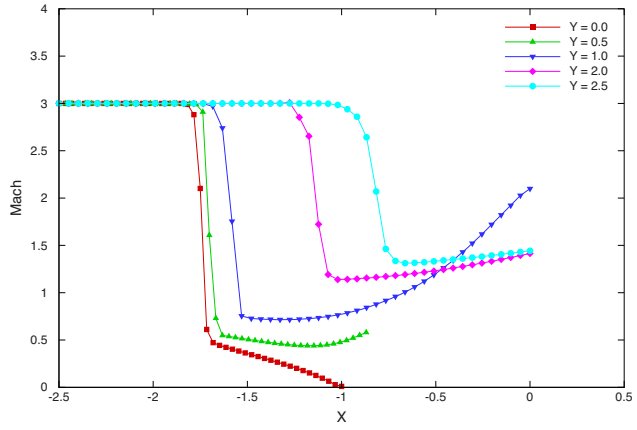



**Fig. 4 2D flow around a cylinder at $M=3$ Mach number. Computed with $b_M=1$, $b_R=1$, and $\rho_{sca}=\rho_2$.**

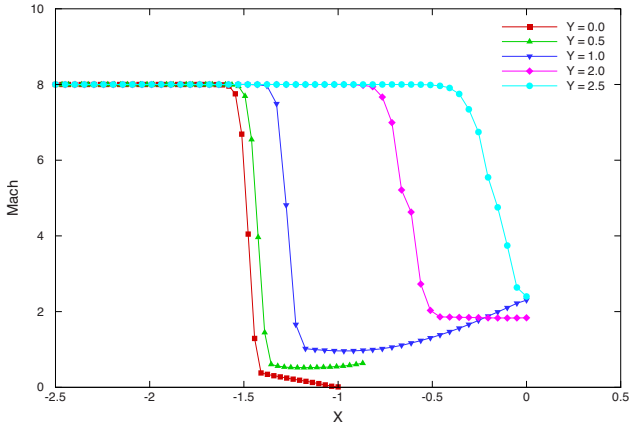



**Fig. 7 2D flow around a cylinder at $M=8$ Mach number. Computed with $b_M=1$, $b_R=1$, and $\rho_{sca}=\rho_2$.**

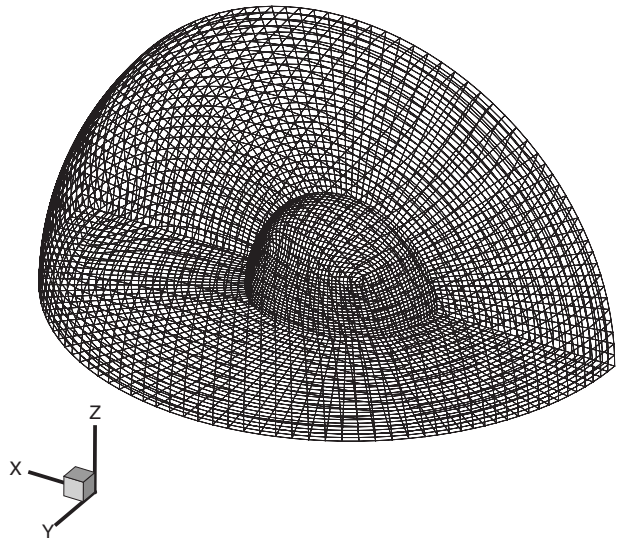



**Fig. 8 3D flow around a sphere. Block-structured mesh with 52,896 hexahedral elements and 56,760 nodes.**

values. On the sphere surface and horizontal boundary, we specify the normal component of the velocity to be zero. No boundary conditions are specified at the outflow boundary.

Figures 9 and 10 show the results obtained with $b_M=1$ and $b_R=0$. Figure 10 represents the only test computation where we activate the parameter $b_F$ by setting it as $b_F=1$. As expected, this results in more dissipation. These results and trends are close to those reported in Ref. [12].

## 6 Concluding Remarks

We carried out an extensive set of numerical experiments with inviscid supersonic flows around cylinders and spheres to evaluate the performance of the $YZ\beta$ shock-capturing combined with the V-SGS method. The $YZ\beta$ shock-capturing technique was introduced originally for use in combination with the SUPG formulation of compressible flows in conservation variables. It is simpler than the shock-capturing parameter $\delta_{91}$, which was derived in 1991 from a shock-capturing parameter given for the entropy variables. The V-SGS method is based on an approximation of the class of SGS models derived from the Hughes-VMS method. As test cases, we computed 2D flow around a cylinder at Mach numbers 3 and 8 and 3D flow around a sphere at Mach number 3 using a structured mesh with quadrilateral elements in 2D and a block-

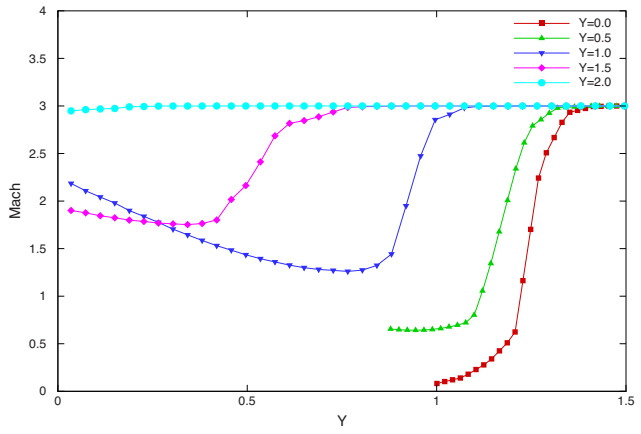



**Fig. 9 3D flow around a sphere at $M=3$ Mach number. Computed with $b_M=1$ and $b_R=0$.**

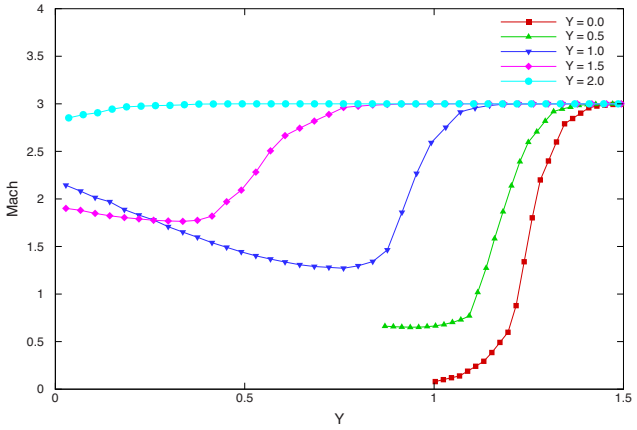



**Fig. 10 3D flow around a sphere at $M=3$ Mach number. Computed with $b_M=1$, $b_R=0$, and $b_F=1$.**

structured mesh with hexahedral elements in 3D. The results and shock qualities obtained from these test computations are very close to those obtained with the $YZ\beta$ shock-capturing combined with the SUPG formulation, which were shown in an earlier article to compare very favorably to the results obtained with the well established OVERFLOW code.

## Acknowledgment

This work was supported by the Department of Mechanics and Aeronautics, University of Rome "La Sapienza" under the Bilateral Agreement UDEP/"La Sapienza." Partial support for this work was provided by the Italian Ministry of University and Academic Research under the Visiting Professor Program, 2006–2007. T.E.T. was supported in part by NASA Johnson Space Center under Grant No. NNJ06HG84G.

## References

[1] Hughes, T. J. R., and Brooks, A. N., 1979, "A Multi-Dimensional Upwind Scheme With No Crosswind Diffusion," *Finite Element Methods for Convection Dominated Flows*, AMD-Vol. 34, T. J. R. Hughes, ed., ASME, New York, pp. 19–35.

[2] Brooks, A. N., and Hughes, T. J. R., 1982, "Streamline Upwind/Petrov-Galerkin Formulations for Convection Dominated Flows With Particular Emphasis on the Incompressible Navier-Stokes Equations," Comput. Methods Appl. Mech. Eng., **32**, pp. 199–259.

[3] Tezduyar, T. E., and Hughes, T. J. R., 1983, "Finite Element Formulations for Convection Dominated Flows With Particular Emphasis on the Compressible Euler Equations," AIAA Paper No. 83-0125.

[4] Hughes, T. J. R., and Tezduyar, T. E., 1984, "Finite Element Methods for First-Order Hyperbolic Systems With Particular Emphasis on the Compressible Euler Equations," Comput. Methods Appl. Mech. Eng., **45**, pp. 217–284.

[5] Hughes, T. J. R., Franca, L. P., and Mallet, M., 1987, "A New Finite Element Formulation for Computational Fluid Dynamics: VI. Convergence Analysis of the Generalized SUPG Formulation for Linear Time-Dependent Multi-Dimensional Advective-Diffusive Systems," Comput. Methods Appl. Mech. Eng., **63**, pp. 97–112.

[6] Le Beau, G. J., and Tezduyar, T. E., 1991, "Finite Element Computation of Compressible Flows With the SUPG Formulation," *Advances in Finite Element Analysis in Fluid Dynamics*, FED-Vol. 123, ASME, New York, pp. 21–27.

[7] Le Beau, G. J., Ray, S. E., Aliabadi, S. K., and Tezduyar, T. E., 1993, "SUPG Finite Element Computation of Compressible Flows With the Entropy and Conservation Variables Formulations," Comput. Methods Appl. Mech. Eng., **104**, pp. 397–422.

[8] Tezduyar, T. E., 2004, "Finite Element Methods for Fluid Dynamics With Moving Boundaries and Interfaces," *Encyclopedia of Computational Mechanics, Volume 3: Fluids*, E. Stein, R. De Borst, and T. J. R. Hughes, eds., Wiley, New York.

[9] Tezduyar, T. E., 2004, "Determination of the Stabilization and Shock-Capturing Parameters in SUPG Formulation of Compressible Flows," *Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2004*, Jyvaskyla, Finland.

[10] Tezduyar, T. E., and Senga, M., 2006, "Stabilization and Shock-Capturing Parameters in SUPG Formulation of Compressible Flows," Comput. Methods Appl. Mech. Eng., **195**, pp. 1621–1632.

[11] Tezduyar, T. E., and Senga, M., 2007, "SUPG Finite Element Computation of

Inviscid Supersonic Flows With YZ$\beta$ Shock-Capturing," Comput. Fluids, **36**, pp. 147–159.
[12] Tezduyar, T. E., Senga, M., and Vicker, D., 2006, "Computation of Inviscid Supersonic Flows Around Cylinders and Spheres With the SUPG Formulation and YZ$\beta$ Shock-Capturing," Comput. Mech., **38**, pp. 469–481.
[13] Buning, P. G., Jespersen, D. C., Pulliam, T. H., Klopfer, G. H., Chan, W. M., Slotnick, J. P., Krist, S. E., and Renze, K. J., 2000, OVERFLOW User's Manual, Version 1.8s, NASA Langley Research Center, Hampton, VA.
[14] Corsini, A., Rispoli, F., and Santoriello, A., 2005, "A Variational Multiscale High-Order Finite Element Formulation for Turbomachinery Flow Computations," Comput. Methods Appl. Mech. Eng., **194**, pp. 4797–4823.
[15] Hughes, T. J. R., 1995, "Multiscale Phenomena: Green's Functions, the Dirichlet-to-Neumann Formulation, Subgrid Scale Models, Bubbles, and the Origins of Stabilized Methods," Comput. Methods Appl. Mech. Eng., **127**, pp. 387–401.
[16] Rispoli, F., and Saavedra, R., 2006, "A Stabilized Finite Element Method Based on SGS Models for Compressible Flows," Comput. Methods Appl. Mech. Eng., **196**, pp. 652–664.
[17] Rispoli, F., Saavedra, R., Corsini, A., and Tezduyar, T. E., 2007, "Computation of Inviscid Compressible Flows With the V-SGS Stabilization and YZ$\beta$ Shock-Capturing," Int. J. Numer. Methods Fluids, **54**, pp. 695–706.
[18] Tezduyar, T. E., and Osawa, Y., 2000, "Finite Element Stabilization Parameters Computed From Element Matrices and Vectors," Comput. Methods Appl. Mech. Eng., **190**, pp. 411–430.

# Time-Derivative Preconditioning Methods for Multicomponent Flows—Part I: Riemann Problems

**Jeffrey A. Housman**
University of California Davis,
2132 Bainer Hall,
One Shields Avenue,
Davis, CA 95616

**Cetin C. Kiris**
NASA Advanced Supercomputing (NAS) Division,
NASA Ames Research Center,
Moffett Field, CA 94035

**Mohamed M. Hafez**
University of California Davis,
2132 Bainer Hall,
One Shields Avenue,
Davis, CA 95616

*A time-derivative preconditioned system of equations suitable for the numerical simulation of inviscid multicomponent and multiphase flows at all speeds is described. The system is shown to be hyperbolic in time and remains well conditioned in the incompressible limit, allowing time marching numerical methods to remain an efficient solution strategy. It is well known that the application of conservative numerical methods to multicomponent flows containing sharp fluid interfaces will generate nonphysical pressure and velocity oscillations across the component interface. These oscillations may lead to stability problems when the interface separates fluids with large density ratio, such as water and air. The effect of which may lead to the requirement of small physical time steps and slow subiteration convergence for implicit time marching numerical methods. At low speeds the use of nonconservative methods may be considered. In this paper a characteristic-based preconditioned nonconservative method is described. This method preserves pressure and velocity equilibrium across fluid interfaces, obtains density ratio independent stability and convergence, and remains well conditioned in the incompressible limit of the equations. To extend the method to transonic and supersonic flows containing shocks, a hybrid formulation is described, which combines a conservative preconditioned Roe method with the nonconservative preconditioned characteristic-based method. The hybrid method retains the pressure and velocity equilibrium at component interfaces and converges to the physically correct weak solution. To demonstrate the effectiveness of the nonconservative and hybrid approaches, a series of one-dimensional multicomponent Riemann problems is solved with each of the methods. The solutions are compared with the exact solution to the Riemann problem, and stability of the numerical methods are discussed.* [DOI: 10.1115/1.3072905]

*Keywords: hybrid conservative/nonconservative method, split coefficient matrix (SCM) method, time-derivative preconditioning methods, dual time stepping*

## 1 Introduction

Many propulsion related flow applications require modeling of multicomponent/multiphase flows over a wide range of Mach numbers, for example, the low speed flow of liquid propellants through the low pressure fuel turbopump (LPFTP) in the space shuttle main engine (SSME), see Ref. [1]. In this case liquid propellant is pumped from low to high pressures where cavitation is likely to occur and may adversely affect the efficiency of the propulsion system. Another example is the overpressure suppression system activated during the launch of a space vehicle, see Ref. [2]. During lift off, initial pressure waves generated at ignition reflect from the ground back to the vehicle, which may effect its stability. The suppression system consists of a water injection system and water trough covers located in the exhaust holes on the launch platform. When the plume of chemically reacting exhaust gases enters the exhaust holes, water is injected and the water baths vaporize to suppress the ignition overpressure phenomenon. The exhaust plume travels at supersonic speeds as it leaves the nozzle while the liquid water jets are nearly incompressible. In order to simulate this complex physical phenomenon, accurate and efficient numerical methods capable of modeling multicomponent/multiphase chemically reacting flow physics must be developed. As a first step toward this goal, a time-derivative preconditioned numerical method is described for the simulation of multicomponent/multiphase inviscid compressible fluids obeying an arbitrary equation of state. The formulation is an extension of the single component compressible formulation presented in Ref. [3] to multicomponent mixtures.

It has been established that time marching numerical methods used to solve the compressible equations become inefficient and lose accuracy when applied to low speed flows [4,5]. It has also been demonstrated that conservative numerical methods applied to multicomponent flows produce nonphysical pressure and velocity oscillations across fluid interfaces [6–10]. In this work a time-derivative preconditioned system of equations for inviscid multicomponent is described along with a characteristic-based nonconservative numerical method, which eliminates the nonphysical behavior across fluid interfaces and remains well conditioned in the low speed limit. The new nonconservative method is an extension of the split coefficient matrix (SCM) method, developed by Chakravarthy et al. [11], to low speed flows utilizing time-derivative preconditioning and a specific set of primitive variables that preserve pressure and velocity equilibrium across component interfaces. Additionally, it is well known that nonconservative methods do not converge to the physically correct weak solution when shock waves are present in the flow field. In order to obtain a numerical method capable of maintaining pressure and velocity equilibrium across component interfaces, to converge to the correct weak solution for high speed flows, and to remain well

Contributed by the Applied Mechanics Division of ASME for publication in the JOURNAL OF APPLIED MECHANICS. Manuscript received January 31, 2008; final manuscript received July 10, 2008; published online February 4, 2009. Review conducted by Tayfun E. Tezduyar.

conditioned in the incompressible limit, the authors have developed a hybrid formulation combining the nonconservative preconditioned SCM method with a conservative preconditioned Roe (PROE) method. The hybrid method (denoted HYBR) utilizes the conservative method almost everywhere and reduces to the nonconservative method only near fluid interfaces such that nonphysical oscillations are not generated. The resulting method is accurate and robust for solving multicomponent flows at all flow speeds. Alternative strategies have been implemented using interface tracking in the finite element framework, see Ref. [12].

Part I of this paper is organized as follows. First, the time-derivative preconditioned system of equations is presented. Next, the numerical discretization of the preconditioned equations is described, including the conservative PROE method, the nonconservative preconditioned split coefficient matrix (PSCM) method, and the hybrid HYBR method. Then, a series of one-dimensional multicomponent Riemann problems is solved using each of the methods. Each of the methods is compared with the exact solution of the Riemann problem, and stability and performance of the methods are also discussed. Application of the methods to multidimensional problems are reported in Part II of this paper.

## 2 Formulation

The concept of time-derivative preconditioning to modify the convergence properties of time marching numerical methods was first developed by Chorin [13] for the steady incompressible Navier–Stokes equations. The artificial compressibility method enabled well-established time marching solution strategies, originally developed for transonic flow calculations, to be applied to steady incompressible flows. These methods were extended to unsteady three-dimensional flows with complex geometries, see Refs. [14–16]. Extension of the artificial compressibility concept to low speed compressible flows started with work on low Mach number perturbation expansions of the compressible equations performed by Rehm and Baum [17] as well as by Klainerman and Majda [18,19]. This perturbation analysis stimulated the generalization of the artificial compressibility method to time-derivative preconditioning methods for compressible flows, which first appeared in literature in Refs. [20,21,4,22,5]. The original motivation of time-derivative preconditioning was to reduce the stiffness associated with characteristic speeds of the compressible equations in the low speed limit. The local preconditioning allowed the time marching method to avoid the deterioration of the convergence rate that nonpreconditioned methods experienced when applied to low speed flows. It was later discovered that the accuracy of the numerical solutions was also improved by low speed preconditioning through the modification of the artificial dissipation present in the numerical fluxes. Application of time-derivative preconditioning methods to multicomponent/multiphase flows is relatively new, see Refs. [23–28]. In the aforementioned references, no discussion of the nonphysical pressure and velocity oscillations generated across a fluid interface by conservative numerical methods is presented. This nonphysical behavior must be included when assessing the validity of a numerical method.

In this section, the time-derivative preconditioned system of equations is described for an inviscid compressible mixture of fluid components, each obeying an arbitrary equation of state. The fluid components may represent different gas species as in multispecies reacting flows or different fluid phases as in multiphase cavitating flows, with the restriction that each component is assumed to maintain an equilibrium pressure, velocity, and temperature. This allows each fluid component to have there own individual densities, enthalpies, entropies, sound speeds, etc. To ensure that the equations remain well conditioned in the low speed limit, time-derivative preconditioning is added to the system, as described in Refs. [3,29].

The time-derivative preconditioned system of equations for an inviscid $N$-component mixture of compressible fluids written in strong conservation law form for a nonorthogonal curvilinear coordinate system $(\xi(x,y,t), \eta(x,y,t)$ is given by

$$\Gamma_p \frac{\partial \hat{Q}}{\partial s} + \frac{\partial \hat{W}}{\partial t} + \frac{\partial \hat{F}}{\partial \xi} + \frac{\partial \hat{G}}{\partial \eta} = 0 \tag{1}$$

where

$$\hat{Q} = J^{-1}\begin{bmatrix} p \\ u \\ v \\ T \\ Y_1 \\ \vdots \\ Y_{N-1} \end{bmatrix}, \quad \hat{W} = J^{-1}\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho H - p \\ \rho Y_1 \\ \vdots \\ \rho Y_{N-1} \end{bmatrix}$$

$$\hat{F} = \begin{bmatrix} \rho\hat{U} \\ \rho\hat{U}u + \hat{\xi}_x p \\ \rho\hat{U}v + \hat{\xi}_y p \\ \rho\hat{U}H - \hat{\xi}_t p \\ \rho\hat{U}Y_1 \\ \vdots \\ \rho\hat{U}Y_{N-1} \end{bmatrix}, \quad \hat{G} = \begin{bmatrix} \rho\hat{V} \\ \rho\hat{V}u + \hat{\eta}_x p \\ \rho\hat{V}v + \hat{\eta}_y p \\ \rho\hat{V}H - \hat{\eta}_t p \\ \rho\hat{V}Y_1 \\ \vdots \\ \rho\hat{V}Y_{N-1} \end{bmatrix} \tag{2}$$

Note that both a preconditioned pseudo-time-derivative in $s$ and physical time derivative in $t$ are included in the formulation such that time accurate flows can be modeled using the dual time stepping method. Standard notation is used for the fluid dynamic variables pressure $p$, velocity components $u$ and $v$, and temperature $T$. The mixture fluid properties are represented by $\rho$ for the mixture density, $H=h+(u^2+v^2)/2$ for the mixture total enthalpy, and $h$ for the mixture specific enthalpy. The mass fraction of the $i$th fluid component is represented by $Y_i$ for $i=1,\ldots,N-1$, where the mass fraction of the $N$th component is given by the saturation condition

$$Y_N = 1 - \sum_{i=1}^{N-1} Y_i \tag{3}$$

The mixture properties are defined using either Amagat's and Dalton's mixture laws, see Ref. [25] for details. These definitions are given as follows:

$$\text{mixture density } \frac{1}{\rho} = \sum_{i=1}^{N} \frac{Y_i}{\rho_i(p,T)}$$

$$\text{mixture internal energy } e = \sum_{i=1}^{N} e_i(p,T) Y_i$$

$$\text{mixture enthalpy } h = \sum_{i=1}^{N} h_i(p,T) Y_i$$

$$\text{mixture entropy } s = \sum_{i=1}^{N} s_i(p,T) Y_i \tag{4}$$

Two thermodynamic quantities are necessary to define the thermodynamic state of an individual component. Since the pressure and temperature are assumed to be in equilibrium for all components they are a natural choice. This choice is implicit in the mixture property definitions given in Eq. (4). To close the system of equa-

tions, an equation of state for each individual component of the fluid is required.

The scaled metric terms are given by

$$\hat{\xi}_x = \xi_x/J = y_\eta, \quad \hat{\xi}_y = \xi_y/J = -x_\eta$$

$$\hat{\xi}_t = \xi_t/J = -x_t\hat{\xi}_x - y_t\hat{\xi}_y$$

$$\hat{\eta}_x = \eta_x/J = -y_\xi, \quad \hat{\eta}_y = \eta_y/J = x_\xi$$

$$\hat{\eta}_t = \eta_t/J = -x_t\hat{\eta}_x - y_t\hat{\eta}_y$$

and the scaled contravariant velocities are

$$\hat{U} = \hat{\xi}_t + u\hat{\xi}_x + v\hat{\xi}_y \quad \text{and} \quad \hat{V} = \hat{\eta}_t + u\hat{\eta}_x + v\hat{\eta}_y$$

where the inverse of the determinant of the Jacobian used in the scaling above is given by

$$J^{-1} = x_\xi y_\eta - x_\eta y_\xi$$

The local time-derivative preconditioning matrix is defined as

$$\Gamma_p = \begin{bmatrix} \rho_p' & 0 & 0 & \rho_T & \rho_{Y_1} & \cdots & \rho_{Y_{N-1}} \\ u\rho_p' & \rho & 0 & u\rho_T & u\rho_{Y_1} & \cdots & u\rho_{Y_{N-1}} \\ v\rho_p' & 0 & \rho & v\rho_T & v\rho_{Y_1} & \cdots & v\rho_{Y_{N-1}} \\ H\rho_p' + \rho h_p - 1 & \rho u & \rho v & H\rho_T + \rho h_T & H\rho_{Y_1} + \rho h_{Y_1} & \cdots & H\rho_{Y_{N-1}} + \rho h_{Y_{N-1}} \\ Y_1\rho_p' & 0 & 0 & Y_1\rho_T & Y_1\rho_{Y_1} + \rho & \cdots & Y_1\rho_{Y_{N-1}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ Y_{N-1}\rho_p' & 0 & 0 & Y_{N-1}\rho_T & Y_{N-1}\rho_{Y_1} & \cdots & Y_{N-1}\rho_{Y_{N-1}} + \rho \end{bmatrix} \tag{5}$$

where the partial derivatives of the material properties $\rho_p, \rho_T, \rho_{Y_1}, \ldots, \rho_{Y_{N-1}}$ and $h_p, h_T, h_{Y_1}, \ldots, h_{Y_{N-1}}$ are defined as

$$\frac{\partial \rho}{\partial p} = \rho^2 \sum_{i=1}^{N} \frac{Y_i}{\rho_i}\frac{\partial \rho_i}{\partial p}, \quad \frac{\partial h}{\partial p} = \sum_{i=1}^{N} Y_i \frac{\partial h_i}{\partial p} \tag{6}$$

$$\frac{\partial \rho}{\partial T} = \rho^2 \sum_{i=1}^{N} \frac{Y_i}{\rho_i}\frac{\partial \rho_i}{\partial T}, \quad \frac{\partial h}{\partial T} = \sum_{i=1}^{N} Y_i \frac{\partial h_i}{\partial T} \tag{7}$$

$$\frac{\partial \rho}{\partial Y_1} = \rho^2\left(\frac{1}{\rho_N} - \frac{1}{\rho_1}\right), \quad \frac{\partial h}{\partial Y_1} = h_1 - h_N \tag{8}$$

$$\vdots \tag{9}$$

$$\frac{\partial \rho}{\partial Y_{N-1}} = \rho^2\left(\frac{1}{\rho_N} - \frac{1}{\rho_{N-1}}\right), \quad \frac{\partial h}{\partial Y_{N-1}} = h_{N-1} - h_N \tag{10}$$

and the local preconditioning parameter $\rho_p'$ is defined by

$$\rho_p' = \frac{1}{V_p^2} - \frac{\rho_T(1-\rho h_p)}{\rho h_T} \tag{11}$$

and the characteristic velocity scale is

$$V_p^2 = \min(c, \max(U, L_{\text{uns}}/\Delta t_{\text{uns}}, \beta)) \tag{12}$$

where $c$ is the isentropic sound speed of the mixture given by

$$c^2 = \frac{\rho h_T}{\rho h_T \rho_p + \rho_T(1-\rho h_p)} \tag{13}$$

the local velocity $U = \sqrt{u^2+v^2}$, $L_{\text{uns}}$ and $\Delta t_{\text{uns}}$ are the reference length and time scales of the unsteady phenomenon, and $\beta > 0$ is a user defined constant that avoids division by zero in the evaluation of $\rho_p'$. Note that as the characteristic velocity scale approaches the physical sound speed the preconditioning system approaches the nonpreconditioned since $\Gamma_p \to \partial W/\partial Q$ as $\rho_p' \to \rho_p$. Derivation of the preconditioned is described in Ref. [3] as well as in Ref. [29]. Additionally, the artificial sound speed, which will appear in the eigenvalue analysis of the preconditioned system, is defined similar to the physical sound speed as

$$c' = \frac{\rho h_T}{\rho h_T \rho_p' + \rho_T(1-\rho h_p)} \tag{14}$$

where $\rho_p$ has been replaced by $\rho_p'$.

The time-derivative preconditioned system of equations defined above is hyperbolic in pseudotime $s$, provided the equations of state are chosen such that the nonpreconditioned system remains hyperbolic in physical time $t$. The eigenvalues of the preconditioned system in the $\xi$-coordinate direction are given by

$$\hat{\lambda}_1 = \frac{1}{2}\left(\hat{U}\left(1+\frac{d}{d'}\right) - \sqrt{\left(\hat{U}\left(1-\frac{d}{d'}\right)\right)^2 + \frac{4\rho h_T}{d'}(\hat{\xi}_x^2 + \hat{\xi}_y^2)}\right)$$

$$\hat{\lambda}_2 = \frac{1}{2}\left(\hat{U}\left(1+\frac{d}{d'}\right) + \sqrt{\left(\hat{U}\left(1-\frac{d}{d'}\right)\right)^2 + \frac{4\rho h_T}{d'}(\hat{\xi}_x^2 + \hat{\xi}_y^2)}\right)$$

$$\hat{\lambda}_3 = \hat{\lambda}_4 = \hat{U} \tag{15}$$

where $d = \rho h_T \rho_p + \rho_T(1-\rho h_p)$ and $d' = \rho h_T \rho_p' + \rho_T(1-\rho h_p)$. Similar relations hold for the eigenvalues in the $\eta$ direction. Note that as $\rho_p'$ approaches the value of the physical material derivative $\rho_p = \partial\rho/\partial p$ then $d' \to d$ and the eigenvalues reduce to the eigenvalues of the nonpreconditioned system. Please refer to Ref. [29] for the explicit form of the eigenvalue/eigenvector decompositions.

## 3 Numerical Method

An efficient time marching numerical method is described for approximating the solution of the time-derivative preconditioned multicomponent flow model described in Sec. 2. For steady state analysis, the physical time-derivative terms are simply omitted from the formulation. Three discretization strategies are outlined for the convective flux derivatives. These include the well known conservative PROE method, the new nonconservative PSCM method, and a conservative/nonconservative hybrid combination of the two methods denoted the HYBR method.

Conservative discretization methods have become the de facto standard for the compressible flow equations. It was shown by Lax and Wendroff [30] that convergent numerical discretizations of hyperbolic equations in discrete conservation law form converge to the correct discontinuous weak solution when shocks are present. Although the use of conservative methods is necessary for

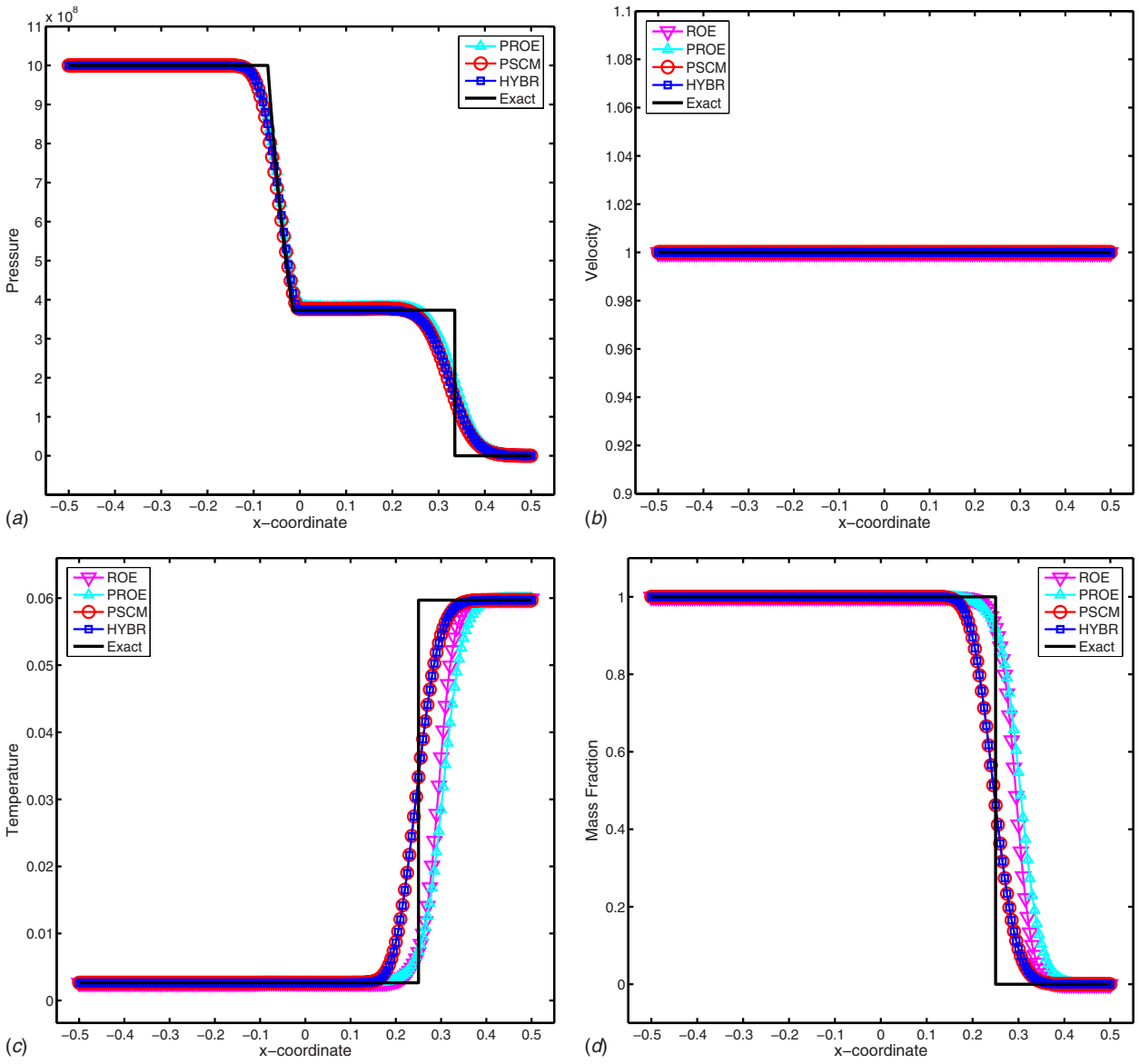



**Fig. 1 Results for Riemann problem I (top to bottom): pressure, velocity, temperature, and mass fraction**

flows containing shocks (unless shock fitting/tracking is used), many low speed compressible flow solutions are free of shocks. Additionally, the use of conservative methods for multicomponent flows generates nonphysical pressure and velocity oscillations across the component interface. For smooth solutions the advantages of a conservative method become questionable, and efficient nonconservative methods should be considered, see, for example, Ref. [31]. When the nonconservative method described below is applied to transonic flows with shocks, the solution does converge to a weak solution of the governing system of equations, but it does not converge to the physically correct entropy satisfying weak solution. In order to model multicomponent flows that contain shock waves, a conservative/nonconservative hybrid method is described.

**3.1 Conservative Formulation.** The conservative PROE method, developed by van Leer et al. [32], is briefly described. First, the governing equations are written in semidiscrete conservative finite difference form as

$$\Gamma_p \frac{\partial \hat{Q}}{\partial s} + \frac{\tilde{F}_{j+1/2} - \tilde{F}_{j-1/2}}{\Delta \xi} + \frac{\tilde{G}_{k+1/2} - \tilde{G}_{k-1/2}}{\Delta \eta} = 0 \tag{16}$$

where $\tilde{F}_{j+1/2}$ and $\tilde{G}_{k+1/2}$ are the numerical fluxes in the $\xi$ and $\eta$ coordinate directions, respectively, while $\Delta\xi$ and $\Delta\eta$ are typically chosen to be one in the computational domain. Note that the physical time derivative has been ignored and will only be considered when the dual time formulation for unsteady analysis is discussed. The preconditioned Roe numerical flux in the $\xi$ direction is given by

$$\tilde{F}_{j+1/2} = \frac{1}{2}[\hat{F}(Q_{j+1}) + \hat{F}(Q_j) - \Gamma_p(\tilde{Q}_{j+1/2})|\Gamma_p^{-1}\hat{A}(\tilde{Q}_{j+1/2})|(Q_{j+1} - Q_j)] \tag{17}$$

The preconditioning matrix $\Gamma_p$ and the absolute value of the preconditioned flux Jacobian $|\Gamma_p^{-1}\hat{A}| = \hat{R}_\xi|\hat{\Lambda}_\xi|\hat{R}_\xi^{-1}$ are evaluated at $\tilde{Q}$, where $\tilde{Q}$ is some symmetric average of the primitive variables $Q_j$

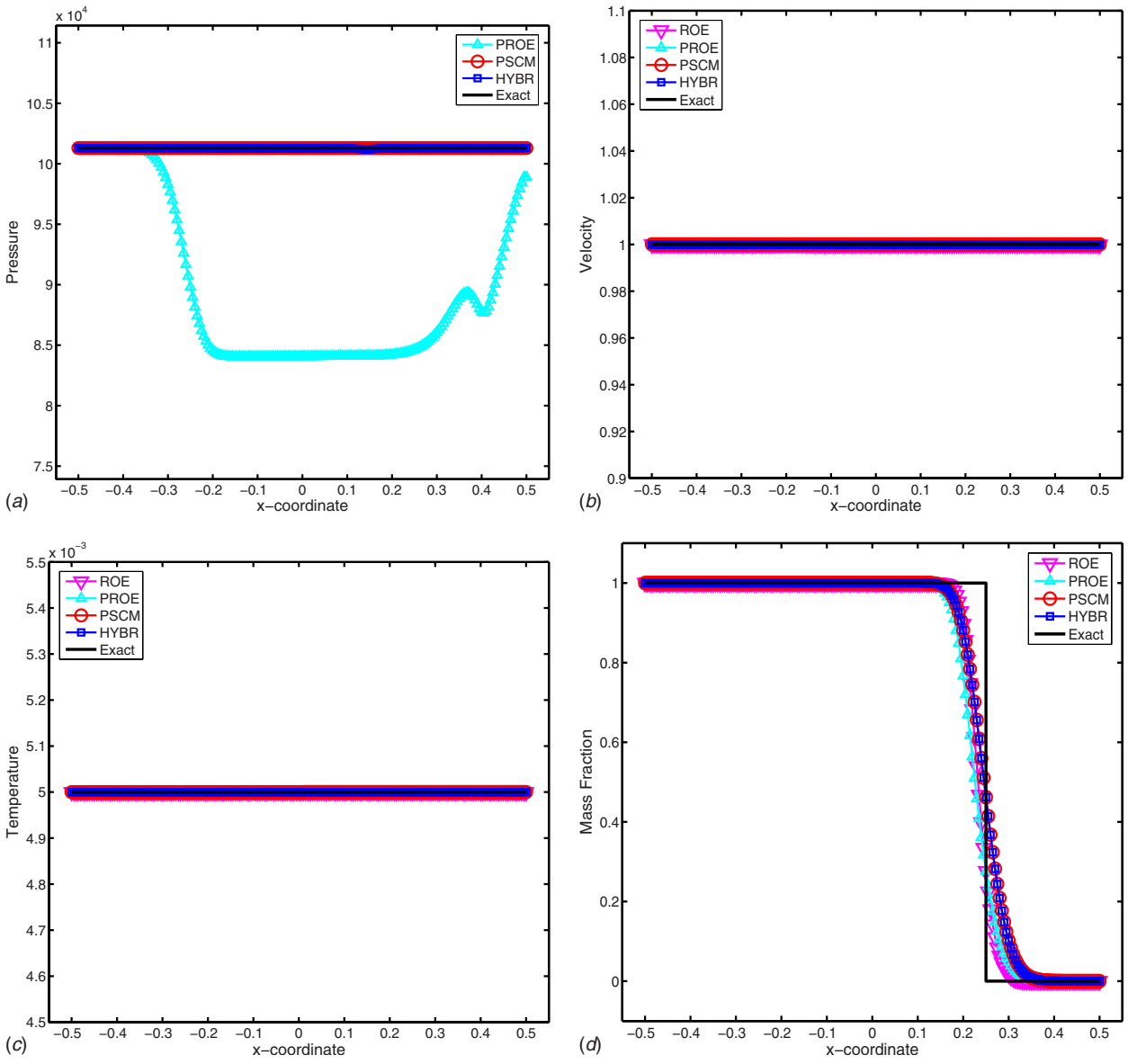



**Fig. 2 Results for Riemann problem II (top to bottom): pressure, velocity, temperature, and mass fraction**

and $Q_{j+1}$. In the present implementation $\tilde{Q}$ takes the form

$$\tilde{Q}_{j+1/2} = \frac{\sqrt{\rho_{j+1}}Q_{j+1} + \sqrt{\rho_j}Q_j}{\sqrt{\rho_{j+1}} + \sqrt{\rho_j}} \tag{18}$$

Although the density weighted average is used, the method described above does not satisfy the discrete Rankine–Hugoniot jump conditions; therefore it is not Roe's method in the strict sense of Roe [33]. The method is conservative and does satisfy all other Roe properties, so weak solutions are correctly captured. Additionally, the arithmetic average $\tilde{Q}=\frac{1}{2}(Q_{j+1}+Q_j)$ was also tested, but the density weighted average proved more robust for multiphase flows with large density ratios.

**3.2 Nonconservative Formulation.** The nonconservative SCM method, developed by Chakravarthy et al. [11], utilizes a characteristic-based splitting similar to the conservative Steger–Warming flux vector splitting method. When the standard SCM method is applied to low speed flows it suffers from the same convergence and accuracy problems as other nonpreconditioned time marching methods. An extension of the SCM method to low speed flows using time-derivative preconditioning and an eigenvalue decomposition based on the primitive variables $Q=(p,u,v,T)^T$ was described in Ref. [3]. This approach is easily extended to the preconditioned multicomponent model described in Sec. 2. By choosing the primitive variables $Q=(p,u,v,T,Y_1,\ldots,Y_{N-1})^T$, pressure and velocity equilibrium across component interfaces is guaranteed; a proof of this is given in Ref. [29]. A brief description of the method is presented below.

To begin, the preconditioned multicomponent flow model can be rewritten in the nonconservative quasilinear form as

$$\frac{\partial \hat{Q}}{\partial s} + \Gamma_p^{-1}\hat{A}\frac{\partial Q}{\partial \xi} + \Gamma_p^{-1}\hat{B}\frac{\partial Q}{\partial \eta} = 0 \tag{19}$$

where $\Gamma_p^{-1}\hat{A}$ and $\Gamma_p^{-1}\hat{B}$ are the preconditioned flux Jacobians, which can be factorized using there eigenvalue decompositions, $\hat{R}_\xi\hat{\Lambda}_\xi\hat{R}_\xi^{-1}$ and $\hat{R}_\eta\hat{\Lambda}_\eta\hat{R}_\eta^{-1}$, respectively. Once factorized the Jacobians are split into their positive and negative eigenvalue contributions, similar to the Steger–Warming splitting but using the primitive variables

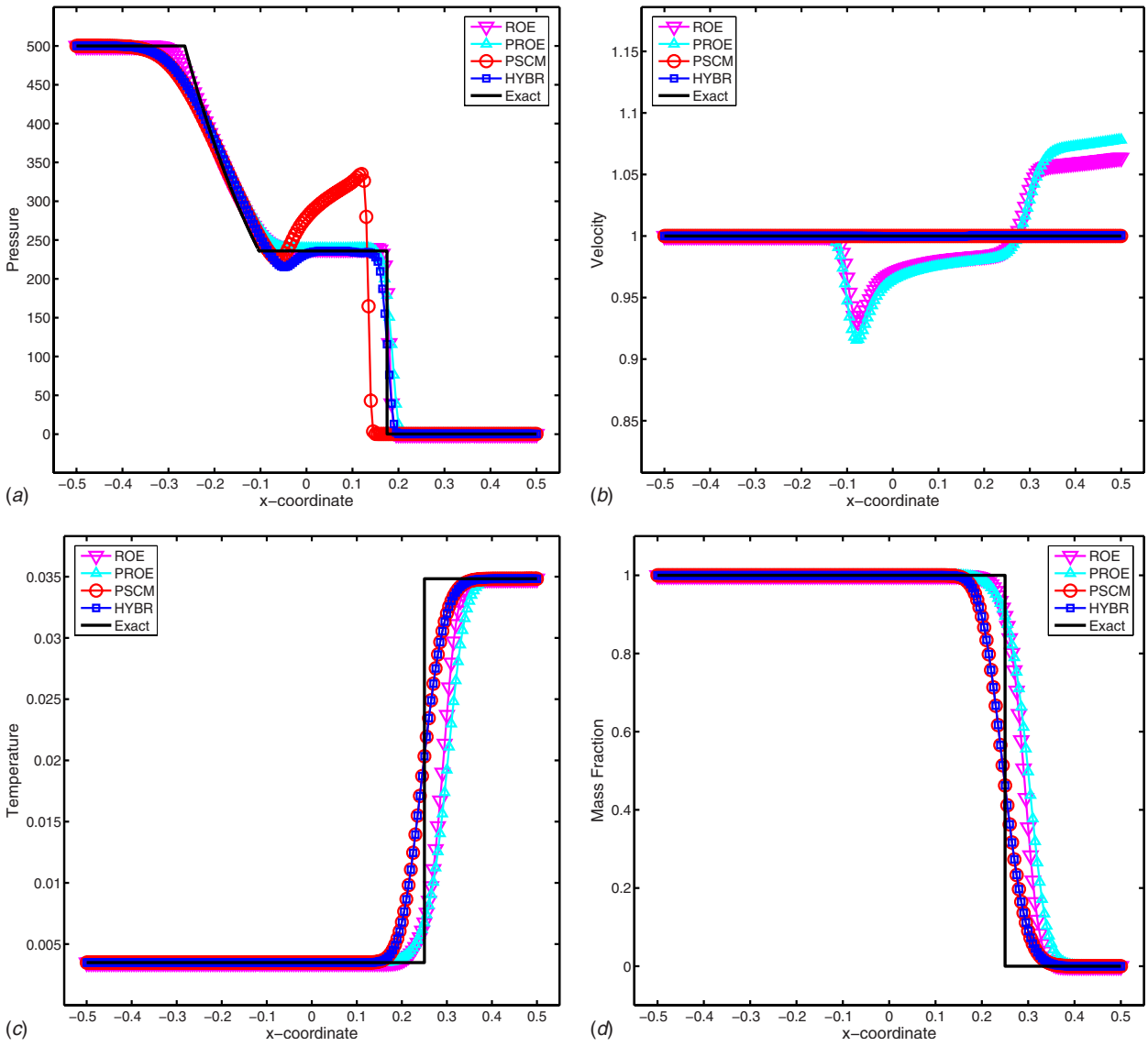



**Fig. 3 Results for Riemann problem III (top to bottom): pressure, velocity, temperature, and mass fraction**

$$\hat{R}_\xi \hat{\Lambda}_\xi \hat{R}_\xi^{-1} = \hat{R}_\xi \hat{\Lambda}_\xi^+ \hat{R}_\xi^{-1} + \hat{R}_\xi \hat{\Lambda}_\xi^- \hat{R}_\xi^{-1}$$

and similarly for the $\eta$ direction flux Jacobian. Note that no assumption of flux homogeneity is required since the method is nonconservative. Define the positive and negative nonconservative flux derivatives as

$$\frac{\partial \hat{F}^\pm}{\partial \xi} \doteq \hat{R}_\xi \hat{\Lambda}_\xi^\pm \hat{R}_\xi^{-1} \frac{\partial Q}{\partial \xi} \quad \text{and} \quad \frac{\partial \hat{G}^\pm}{\partial \eta} \doteq \hat{R}_\eta \hat{\Lambda}_\eta^\pm \hat{R}_\eta^{-1} \frac{\partial Q}{\partial \eta}$$

The split equations are rewritten as

$$\frac{\partial \hat{Q}}{\partial s} + \frac{\partial \hat{F}^+}{\partial \xi} + \frac{\partial \hat{F}^-}{\partial \xi} + \frac{\partial \hat{G}^+}{\partial \eta} + \frac{\partial \hat{G}^-}{\partial \eta} = 0 \tag{20}$$

Note that all of the algebraic manipulations have been performed on the continuous differential model. In the split form, the positive/negative flux derivatives can be discretized using upwind biased differencing. In the original SCM method the eigenvalue decompositions were evaluated at the grid points. Later, Lombard et al. [34] showed that by evaluating the eigenvalue decomposition using an upwind biased average, certain properties (such as conservation in the case of conservative supra-characteristic method (CSCM)) can be obtained. Here the symmetric density weighted average, $\tilde{Q}$, as described above, is used to evaluate the eigenvalue decomposition. Although this choice does not lead to a conservative method, it does result in a robust method for large density ratio flows. The first order upwind difference approximation to the positive/negative flux derivative in the $\xi$ direction is given by

$$\left(\frac{\partial \hat{F}^+}{\partial \xi}\right)_{j-1/2} \approx \hat{R}_\xi(\tilde{Q}_{j-1/2}) \hat{\Lambda}_\xi^+(\tilde{Q}_{j-1/2}) \hat{R}_\xi^{-1}(\tilde{Q}_{j-1/2}) \left[\frac{Q_j - Q_{j-1}}{\Delta \xi}\right]$$

and

$$\left(\frac{\partial \hat{F}^-}{\partial \xi}\right)_{j+1/2} \approx \hat{R}_\xi(\tilde{Q}_{j+1/2}) \hat{\Lambda}_\xi^-(\tilde{Q}_{j+1/2}) \hat{R}_\xi^{-1}(\tilde{Q}_{j+1/2}) \left[\frac{Q_{j+1} - Q_j}{\Delta \xi}\right]$$

For higher order spatial discretizations including limiters the method outlined in Ref. [34] is used.

**3.3 Hybrid Formulation.** Hybrid conservative/nonconservative methods are not new; they were originally developed for single component flows in which the more efficient nonconservative method is used away from shocks and a more computationally expensive conservative method is used in the vi-

cinity of shock discontinuities. For example, Daywitt et al. [35] used the SCM method for steady supersonic flow calculations around blunt bodies where shock fitting was used for the external shock and a local switch to a high-resolution conservative scheme near embedded shocks. Harabetian and Pego [36] developed a hybrid method for the unsteady Euler equations and demonstrated its efficiency over using a strictly conservative method throughout the domain. Similar hybrid methods have been developed by Toro [37], and the hybrid concept has even been used to construct conservative/nonconservative methods known as adaptive Riemann solvers, which switch from one particular approximate Riemann solver to another depending on the local flow features, see Ref. [38]. Hybrid methods applied to multicomponent flows include the methods developed by Karni [8] and Ivings et al. [39]. Here a new time-derivative preconditioned hybrid method is developed, which utilizes both the PROE and PSCM methods described above. To begin, the switching strategy is discussed and a simple switching function that takes on a value of zero near sharp component interfaces and a value of 1 away from the interface. This means that either the conservative method or the nonconservative method is used at any particular grid point, and not a combination of the two, which could lead to consistency problems. Following Ref. [8], the local changes in the mass fraction variable are used to determine if a sharp interface between two components is present in the numerical solution. The switching function takes the following form:

$$\phi_{jk} = \begin{cases} 0 & \text{if } |Y_{j+1} - Y_j| > \varepsilon_Y \\ 0 & \text{if } |Y_j - Y_{j-1}| > \varepsilon_Y \\ 0 & \text{if } |Y_{k+1} - Y_k| > \varepsilon_Y \\ 0 & \text{if } |Y_k - Y_{k-1}| > \varepsilon_Y \\ 1 & \text{otherwise} \end{cases} \tag{21}$$

where $\varepsilon_Y$ is a user defined constant. Most of the test cases performed in the report use a value of $\varepsilon_Y = 1.0 \times 10^{-2}$. When high-resolution methods are used, the stencil of the differences used to check for component interfaces is increased to match the stencil of the unlimited numerical method.

Using the switching function defined above the semidiscrete form of the preconditioned hybrid method is given by

$$\Gamma_p \frac{\partial \hat{Q}}{\partial s} + \phi \left[ \frac{\tilde{F}_{j+1/2} - \tilde{F}_{j-1/2}}{\Delta \xi} + \frac{\tilde{G}_{k+1/2} - \tilde{G}_{k-1/2}}{\Delta \eta} \right]$$
$$+ (1-\phi)\Gamma_p \left[ (\Gamma_p^{-1}\hat{A})^+_{j-1/2} \left( \frac{Q_j - Q_{j-1}}{\Delta \xi} \right) + (\Gamma_p^{-1}\hat{A})^-_{j+1/2} \left( \frac{Q_{j+1} - Q_j}{\Delta \xi} \right) \right]$$
$$+ (1-\phi)\Gamma_p \left[ (\Gamma_p^{-1}\hat{B})^+_{k-1/2} \left( \frac{Q_k - Q_{k-1}}{\Delta \eta} \right) + (\Gamma_p^{-1}\hat{B})^-_{k+1/2} \left( \frac{Q_{k+1} - Q_k}{\Delta \eta} \right) \right] \tag{22}$$

Note that the preconditioning matrix $\Gamma_p$ premultiplying the pseudo-time-derivative as well as the nonconservative flux differences is evaluated at the grid points, while the preconditioned flux Jacobians are evaluated at the cell interfaces for both the conservative and nonconservative methods. Provided the parameter $\varepsilon_Y$ is chosen sufficiently, pressure and velocity equilibrium is maintained across fluid interfaces, and the correct discontinuous weak solution is predicted as the grid is refined, even for strong shocks. Note that for single component flows and/or sufficiently well mixed multicomponent flows the nonconservative method is never turned on and the hybrid method reduces to the preconditioned Roe scheme everywhere.

The use of time-derivative preconditioning destroys the time accuracy of the governing equations. In order to overcome these difficulties the dual time formulation is utilized, as described in Refs. [16,15]. With the dual time approach, the unsteady equations are embedded in a pseudo-time-process and the physical time derivative is discretized using an unconditionally stable (linearly) implicit method, such as implicit Euler or BDF2. A complete description of the dual time discretization for the conservative, nonconservative, and hybrid approaches is given in Ref. [29].

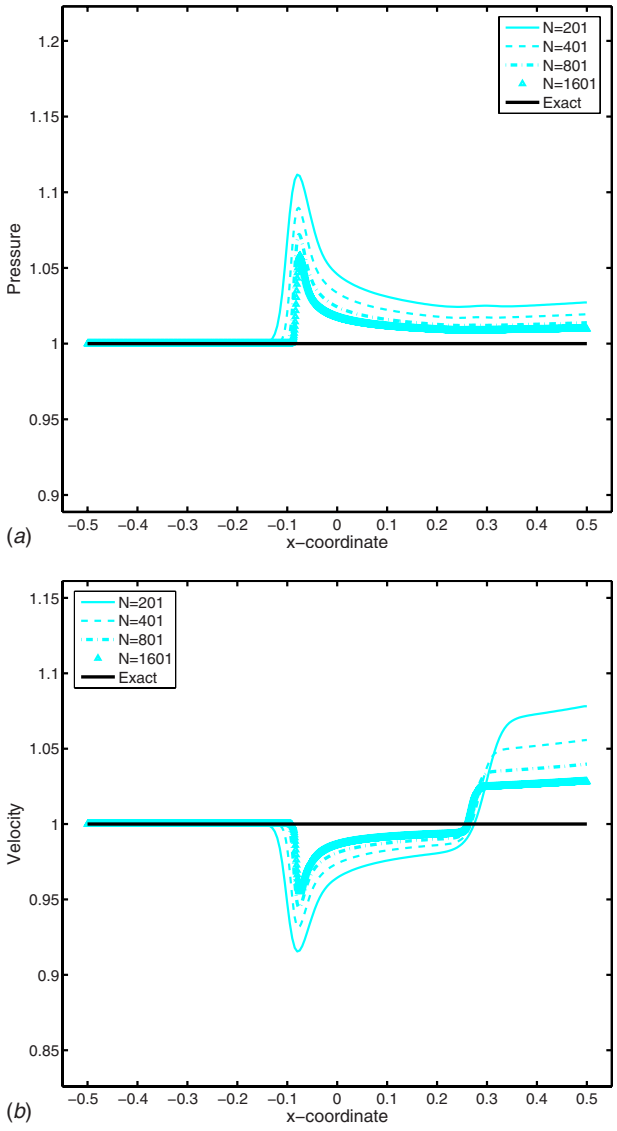



**Fig. 4 Grid convergence of pressure (upper) and velocity (lower) for Riemann problem III using PROE**

Discretization of the spatial and physical time derivatives results in a large system of ordinary equations in pseudotime $s$. For steady state analysis this system is marched in pseudotime to an asymptotic steady state. Upon which the converged solution represents the numerical approximation to the steady governing equations. For unsteady analysis, the system is marched to an asymptotic steady state at each physical time step. Since the pseudotime accuracy of the solution is not important, the discretization method applied to the pseudo-time-derivative is chosen based on stability properties. The implicit Euler time discretization provides a simple and unconditionally stable (linearly) method allowing large pseudo-time-steps to be chosen to accelerate the convergence to an asymptotic steady state.

The system of equations can be written in functional notation as $\partial \hat{Q}/\partial s = \hat{R}(Q)$, where $\hat{R}$ represents the discretized spatial derivatives as well as the discrete physical time derivatives if unsteady analysis is being considered. Applying the implicit Euler time discretization to the semidiscrete system and locally linearizing

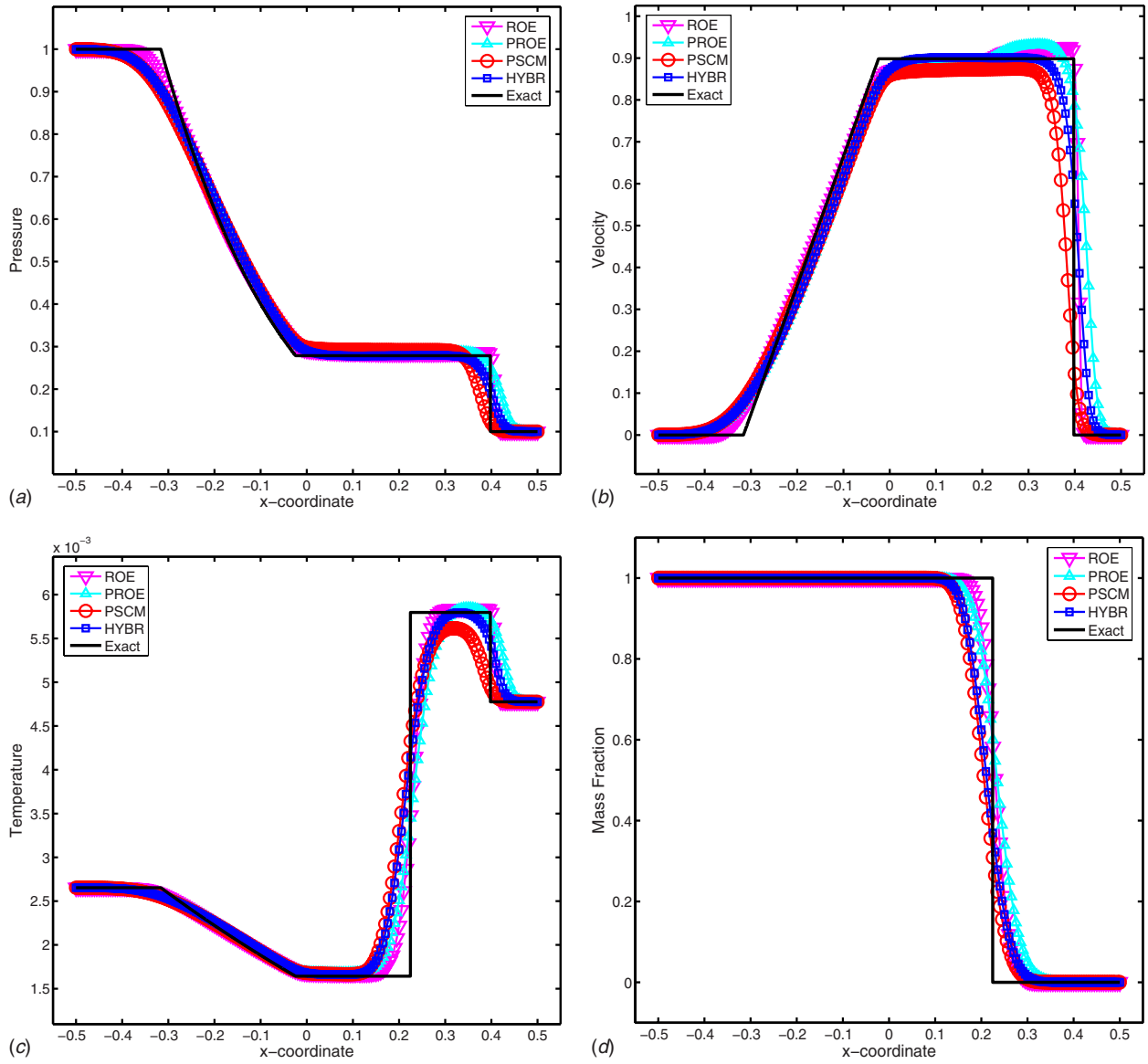



**Fig. 5 Results for Riemann problem IV (top to bottom): pressure, velocity, temperature, and mass fraction**

the resulting nonlinear system of equations about $Q^m$, the *delta form* of the equations is obtained

$$\left[\frac{I}{J\Delta s} - \left(\frac{\partial \hat{R}}{\partial Q}\right)^m\right]\Delta Q^m = \hat{R}(Q^m) \tag{23}$$

where $\partial\hat{R}/\partial Q$ is the system Jacobian of the discrete nonlinear equations. When high-order upwind biased differencing is used, $\partial\hat{R}/\partial Q$ is approximated by the system Jacobian of the first order upwind spatial discretization. This decreases the storage and increases the diagonal dominance of the system allowing efficient relaxation techniques to be used to approximate the solution of the linearized system. For the computed results in this paper an alternating line-implicit Jacobi procedure is used. Typically two to three relaxations suffice to obtain good nonlinear convergence. For a more complete description of the solution procedure see Ref. [29].

## 4 Computed Results

The time-derivative preconditioned system of equations described in Sec. 2 and discretized in Sec. 3 is used to solve a series of one-dimensional multicomponent Riemann problems. Three convective flux derivative discretizations are compared for each of the test cases. These methods include the conservative PROE method, the nonconservative PSCM method, and the conservative/nonconservative HYBR method, which combines the PROE and PSCM methods. Additionally, a nonpreconditioned conservative Roe scheme extended to multicomponent flows, see Ref. [7], is also applied to each of the ideal gas test cases. In each of the cases first order accurate upwind spatial derivatives and implicit Euler time discretization are used. By restricting the order of accuracy the true dissipative nature of the schemes as well as nonphysical behavior can be assessed.

The following series of one-dimensional multicomponent Riemann problems illustrates the difficulties encountered when extending well-established single component numerical methods to multicomponent flows as well as demonstrates the capability of the HYBR method to handle these difficulties. An exact solution of the multicomponent Riemann problem has been derived by the present author for comparison purposes. This solution matches the one reported in Ref. [40]. Unless specified otherwise, a uniform grid of 200 points, a physical time $CFL=\Delta t(|u|+c)/\Delta x=0.9$, and subiteration convergence criteria of five orders of magnitude re-

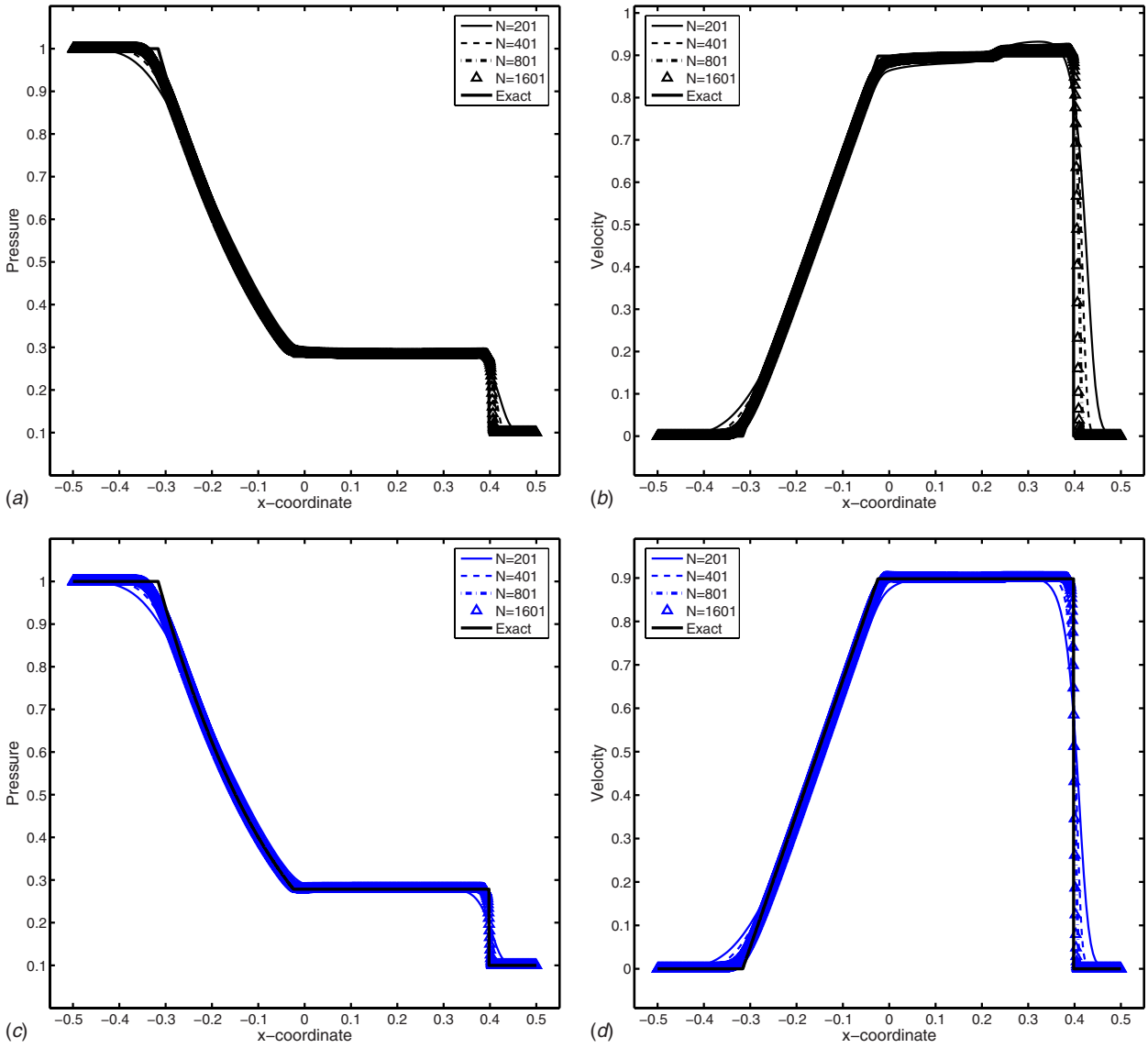



**Fig. 6 Top two: grid convergence of pressure (upper) and velocity (lower) for Riemann problem IV using PROE. Bottom two: grid convergence of pressure (upper) and velocity (lower) for Riemann problem IV using HYBR.**

duction in the maximum residual are used for each of the methods. For the hybrid method, a mass fraction switching value of $\varepsilon_Y = 1.0 \times 10^{-2}$ is used throughout.

**4.1 Riemann Problem I.** The following problem is essentially a single component case solved to ensure that a consistent and accurate numerical implementation of each of the methods has been performed. The initial conditions of the Riemann problem correspond to a moving contact discontinuity, which propagates from left to right.

$$(\rho_L, u_L, p_L, Y_L, \gamma_L, C_{P,L}, P_{\infty,L})^T = (1.0, 1.0, 1.0, 1.0, 1.4, 1005.0, 0.0)^T$$

$$(\rho_R, u_R, p_R, Y_R, \gamma_R, C_{P,R}, P_{\infty,R})^T = (0.1, 1.0, 1.0, 0.0, 1.4, 1005.0, 0.0)^T$$

Figure 1 shows the computed pressure, velocity, temperature, and mass fraction at $t = 0.25$ s. Each of the methods produces oscillation free pressure and velocity profiles as predicted. The conservative methods appear to produce solutions containing a phase error in the temperature and mass fraction variables. If the conserved quantities, such as mass (density), are plotted instead, then this phase error is not present in the solutions produced by the conservative methods, but instead appears in the solutions given by the nonconservative and hybrid methods. This simply has to do with the set of variables being differenced in the physical time derivative. Note that it does not depend on the variables that a particular scheme updates, such as the PROE method, which is conservative but updates the primitive variables.

**4.2 Riemann Problem II.** The next Riemann problem illustrates the fact that if the temperature across the component interface is equal, then the conservative methods do not produce the nonphysical pressure oscillations. This was pointed out by Jenny et al. [9], and often leads to confusion when a conservative method is applied to the moving contact problem. For instance, some authors may claim that a particular conservative method does not generate nonphysical pressure oscillations across the moving contact, but then only demonstrate this when the temperature is equal on both sides of the contact. Here we show that the conservative ROE and PROE methods are able to preserve pressure and velocity equilibrium across the moving contact when the temperature is constant. In the next case, the more general moving contact with different temperatures across the fluid interface is examined.

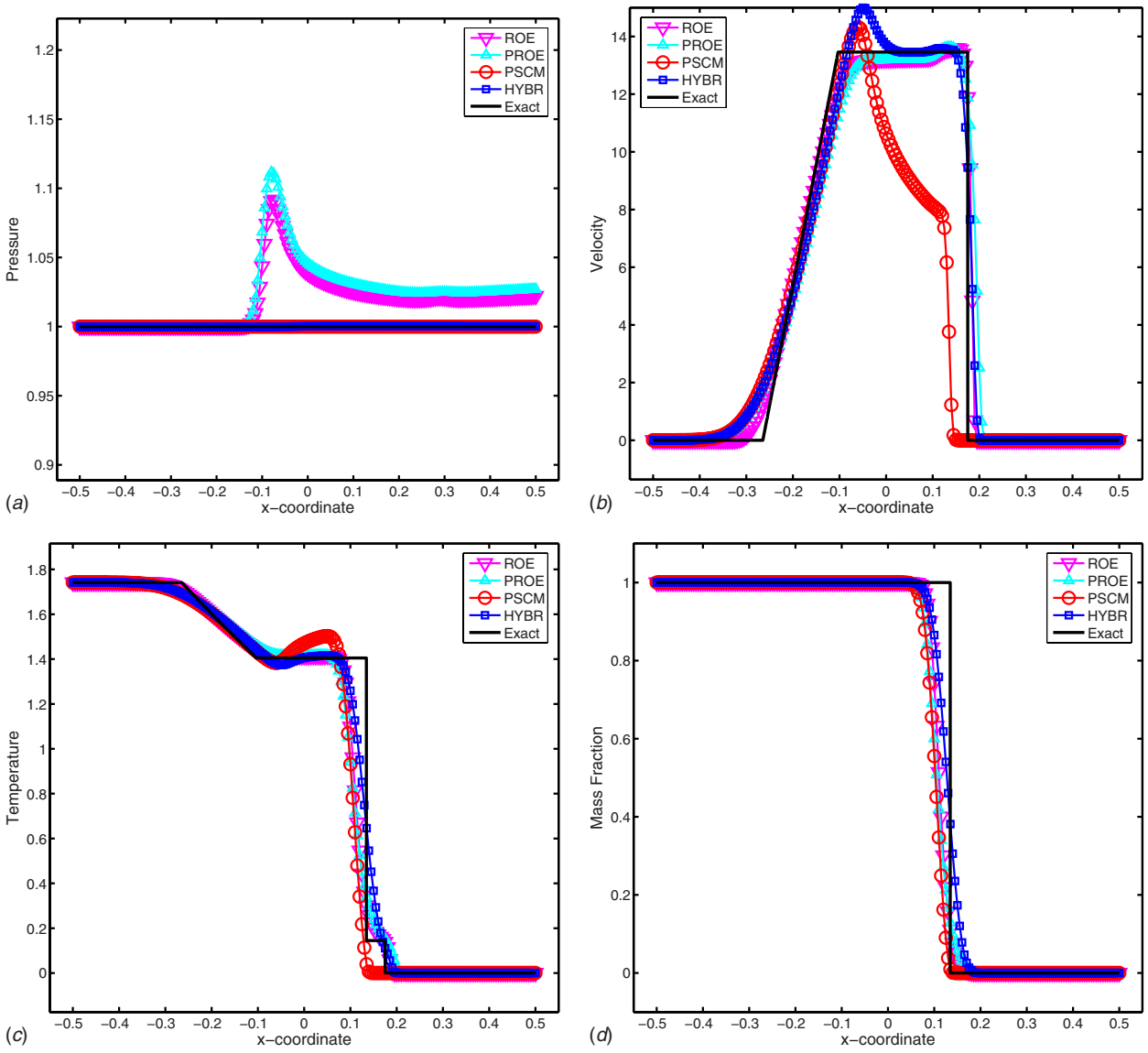



**Fig. 7 Results for Riemann problem V (top to bottom): pressure, velocity, temperature, and mass fraction**

$$(\rho_L, u_L, p_L, Y_L, \gamma_L, C_{P,L}, P_{\infty,L})^T = (0.531, 1.0, 1.0, 1.0, 1.6, 1005.0, 0.0)^T$$

$$(\rho_R, u_R, p_R, Y_R, \gamma_R, C_{P,R}, P_{\infty,R})^T = (1.194, 1.0, 1.0, 0.0, 1.2, 1005.0, 0.0)^T$$

Figure 2 displays the computed pressure, velocity, temperature, and mass fraction at $t=0.25$ s. As predicted in the analysis given by Jenny et al., any conservative method that coincides with the exact Godunov method for the moving contact discontinuity will preserve pressure equilibrium across a gas/gas interface when the temperature is constant across the interface. This does not imply that the conservative methods preserve pressure equilibrium across a moving contact for the more general Riemann problem, as will be demonstrated next.

**4.3 Riemann Problem III.** As alluded to in the previous case, the conservative methods have been predicted to produce nonphysical pressure and velocity oscillations across a moving gas/gas contact when both $\gamma$ and temperature vary across the discontinuity. The purpose of the test case is to demonstrate this fact as well as to show that the nonconservative and hybrid methods accurately predict the contact wave while discretely preserving pressure and velocity equilibrium. The initial data for the case are given below.

$$(\rho_L, u_L, p_L, Y_L, \gamma_L, C_{P,L}, P_{\infty,L})^T = (1.0, 1.0, 1.0, 1.0, 1.6, 1005.0, 0.0)^T$$

$$(\rho_R, u_R, p_R, Y_R, \gamma_R, C_{P,R}, P_{\infty,R})^T = (0.1, 1.0, 1.0, 0.0, 1.2, 1005.0, 0.0)^T$$

Figure 3 plots the pressure, velocity, temperature, and mass fraction at $t=0.25$ s. The nonphysical pressure and velocity oscillations are clearly seen in the solutions produced by the conservative methods, while oscillation free profiles are computed with the nonconservative and hybrid methods. These oscillations are present in the first order method and therefore cannot be suppressed using TVD strategies. It is also necessary to point out that the correct weak solution is obtained by the nonconservative method even though the solution is not smooth. This is because the discontinuity in the solution is associated with the linearly degenerate eigenspace, which moves at the local fluid velocity. Since the nonconservative method also moves this discontinuity at the local fluid velocity the solution coincides with the correct weak solution.

It is often argued that the nonphysical oscillations are not that important and can be reduced through grid refinement. This is a very dangerous argument to use when considering reactive flows where false chemical reactions can take place caused by the non-

physical pressure oscillations. These chemical reactions completely change the character of the computed solution, which may trigger more spurious phenomenon. As for the matter of suppressing the oscillations with grid refinement, we demonstrate in Fig. 4 that even with a mesh spacing of $6.25\times10^{-4}$ over a unit interval the pressure and velocity oscillations are still clearly present.

**4.4 Riemann Problem IV.** In the previous case we established the mechanism for which nonphysical solution behavior is generated when conservative numerical methods are applied to multicomponent problems. The present test case extends this knowledge to the less trivial Riemann problem containing a left moving rarefaction, right moving contact, and right moving shock.

$$(\rho_L,u_L,p_L,Y_L,\gamma_L,C_{P,L},P_{\infty,L})^T=(1.0,0.0,1.0,1.0,1.6,1005.0,0.0)^T$$

$$(\rho_R,u_R,p_R,Y_R,\gamma_R,C_{P,R},P_{\infty,R})^T=(0.125,0.0,0.1,0.0,1.2,1005.0,0.0)^T$$

The initial data given above correspond to a weak shock, but even for this case the nonconservative PSCM method clearly fails to compute the correct shock jumps and appears to have a reduced shock speed. This is displayed in Fig. 5 where the computed pressure, velocity, temperature, and mass fraction are plotted. Additionally, it is observed that the conservative ROE and PROE methods fail to produce the correct velocity jump on the right side of the contact discontinuity. This spurious behavior is more easily observed in Fig. 6 (top two figures) where even under mesh refinement the nonphysical jump persists. The HYBR method clearly performs well on this case and converges to the correct weak solution as the grid is refined, as shown in Fig. 6 (bottom two figures).

**4.5 Riemann Problem V.** It has been established that the hybrid method is able to obtain the correct weak solution for weak shocks, while preserving pressure and velocity equilibrium across fluid interfaces. This next test involves a very strong shock and proves to be a difficult test for all the methods considered here. The initial data, given below, correspond to a left rarefaction, a right moving contact, and a right moving shock.

$$(\rho_L,u_L,p_L,Y_L,\gamma_L,C_{P,L},P_{\infty,L})^T=(1.0,0.0,500.0,1.0,1.6,1005.0,0.0)^T$$

$$(\rho_R,u_R,p_R,Y_R,\gamma_R,C_{P,R},P_{\infty,R})^T=(0.125,0.0,0.2,0.0,1.2,1005.0,0.0)^T$$

Figure 7 shows the computed pressure, velocity, temperature, and mass fraction at $t=0.01$ s. The nonconservative PSCM method appears to give wildly incorrect results, while the conservative methods produce a similar nonphysical velocity jump to the right of the contact, as observed in the previous case. At this grid resolution the HYBR method appears to have a chosen some of the poor features of each of the two underlying methods. This is especially seen in the velocity and pressure profiles. As the grid is refined the nonphysical features are suppressed, as shown in Fig. 8, but it appears that the hybrid method does not handle strong shock wave phenomenon very well. Since we are mostly interested in low speed applications such as cavitation and liquid/solid combustion the present result does not really affect the merits of the HYBR method. With that said, anyone planning to use the HYBR method for high speed reacting flows such as nonequilibrium hypersonic flow may need to modify the components of the scheme considerably. This is not surprising since the standard Roe method has known failures when applied hypersonic flows.

**4.6 Riemann Problem VI.** The previous multicomponent Riemann problems consisted of two ideal gases. The following two problems extend to liquid/gas flows with large density ratios and drastic changes in the equation of state. To begin, the moving contact problem is revisited where the contact now separates water and air, and the initial data for the problem are given below. Note that we have chosen the initial data such that the temperature is not equal across the contact. This case is similar to the one solved by Neaves and Edwards [27] using the *low diffusion flux splitting scheme*, a variant of AUSM. But there initial data assume a constant temperature across the contact; therefore no spurious oscillations were observed.

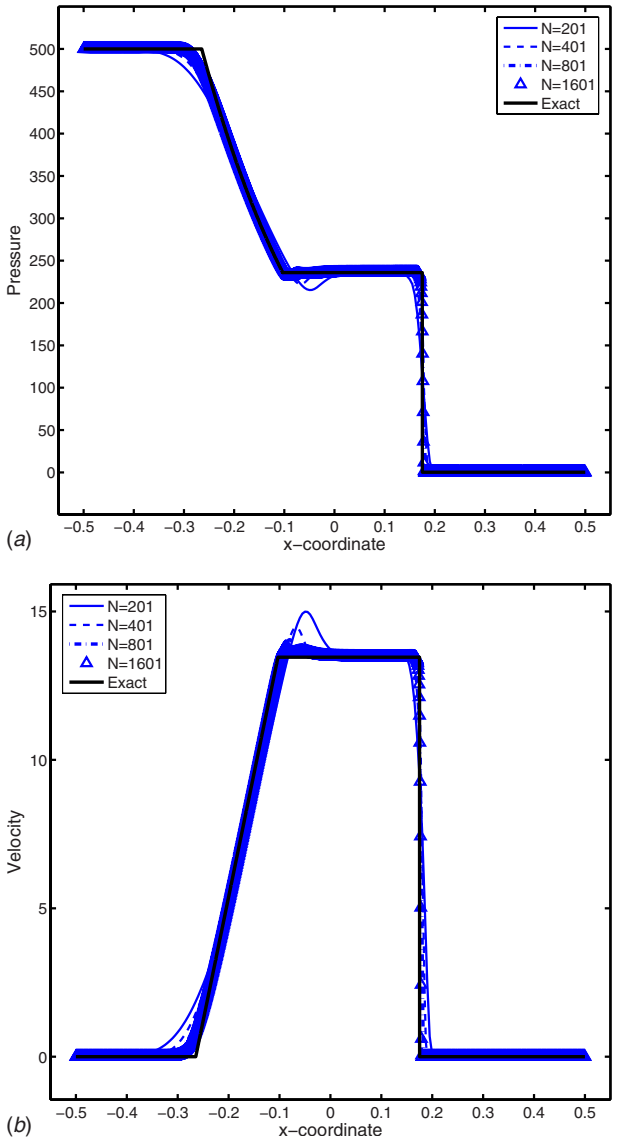



**Fig. 8 Grid convergence of pressure (upper) and velocity (lower) for Riemann problem V using HYBR**

$$(\rho_L,u_L,p_L,Y_L,\gamma_L,C_{P,L},P_{\infty,L})^T$$
$$=(1250.0,675.0,101,300.0,0.0,1.93,8076.7,1.14\times10^{-9})^T$$

$$(\rho_R,u_R,p_R,Y_R,\gamma_R,C_{P,R},P_{\infty,R})^T$$
$$=(1.25,675.0,101,300.0,1.0,1.4,1005.0,0.0)^T$$

Figure 9 displays the computed pressure, velocity, temperature, and mass fraction at $t=4.0\times10^{-4}$ s. It is obvious from the profiles that the conservative method produces large pressure and velocity oscillations on the order of $1.5\times10^4$ Pa and 35 m/s, respectively. Additionally, a physical time $CFL=0.002$ was required by the PROE method such that negative pressures and temperatures were not generated during the time integration, while the PSCM and HYBR methods were run at the efficient physical time $CFL=0.9$ with no problems. Note that $\varepsilon_Y$ was reduced to $1.0\times10^{-3}$ for this test case since some small oscillations in the solution do appear when $\varepsilon_Y=1.0\times10^{-2}$.

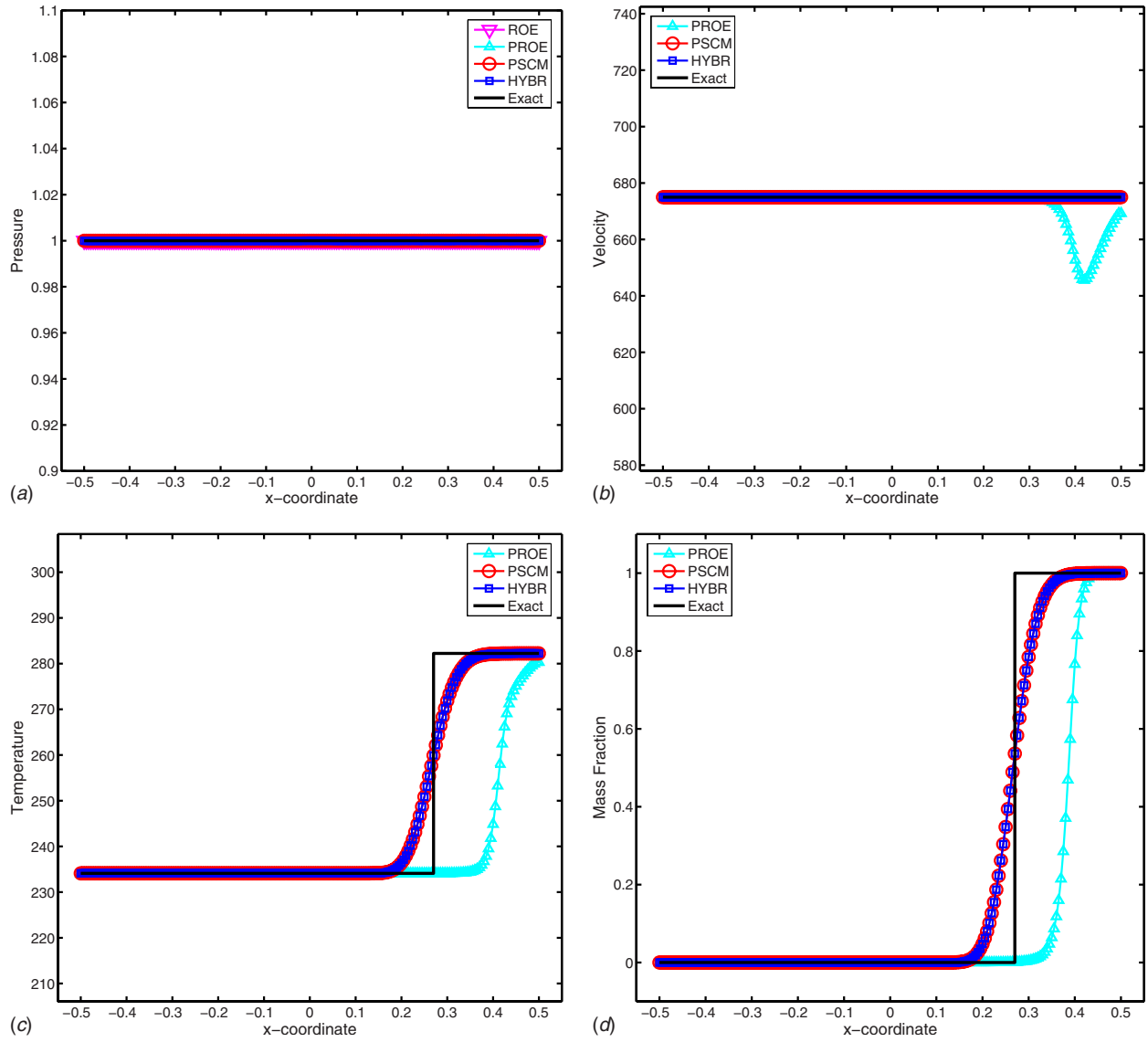



**Fig. 9 Results for Riemann problem VI (top to bottom): pressure, velocity, temperature, and mass fraction**

This case was also ran with constant temperatures across the interface as in the original problem posed by Neaves and Edwards [27]. And similar to the results of the gas/gas problem, no oscillations are generated, but the time step restriction is still present. It is mentioned by Neaves and Edwards that small physical time steps were used for there problems to *minimize time-step errors*, but this may have more to do with stability restrictions caused by the conservative numerical method. Since this version of AUSM has not implemented, there is no way to confirm this. But for the conservative PROE method, severe stability restrictions have been observed.

**4.7 Riemann Problem VII.** The final Riemann problem of the series involves high pressure air shock transmitting through water. The initial data, given below, correspond to a left rarefaction in air, a right moving contact separating air and water, and a right moving shock in water. This case involves large density ratios, varying equations of state, and shock waves moving through liquids.

$$(\rho_L, u_L, p_L, Y_L, \gamma_L, C_{P,L}, P_{\infty,L})^T = (1.16 \times 10^4, 0.0, 1.0 \times 10^9, 1.0, 1.4, 10{,}005.0, 0.0)^T$$

$$(\rho_R, u_R, p_R, Y_R, \gamma_R, C_{P,R}, P_{\infty,R})^T = (9.75 \times 10^2, 0.0, 1.0 \times 10^5, 0.0, 1.92, 8076.7, 1.14 \times 10^9)^T$$

The initial temperature profile for this case is constant and the shock wave is weak so all three methods appear to perform adequately. Note that the physical time *CFL* was reduced to 0.8 for the PROE methods such that negative pressure nor temperature was generated during the time integration process. It is interesting that all of the methods excessively smear the shock. This is not observed when the shock moves through a gas, and must have something to do with the stiffened gas equation of state. This implies that high fidelity grid resolution is necessary when modeling shocks moving through liquids using the current methods described.

Concluding the series of multicomponent Riemann problems, we have established and demonstrated the nonphysical behavior of conservative multicomponent methods, observed severe physical time step stability restrictions using the PROE method when liquid/gas flows are being modeled, and have shown that the nonphysical oscillations, although are reduced, still exist as the grid is

refined. Alternatively, we have shown that the PSCM method does not suffer from the nonphysical pressure and velocity oscillations, converges to the correct weak solution when discontinuities only exist in the linearly degenerate eigenspaces, but does not converge to the correct weak solutions when shocks are present. Additionally, the combination of the two methods in the HYBR method is shown to preserve pressure and velocity equilibrium across contacts, converges to the correct weak solution (even for strong shocks provided the grid is well resolved), and does not suffer from any time step stability restrictions when liquid/gas flows are being considered. Therefore the HYBR method appears to be the preferred method for multicomponent flows. The remainder of the chapter is designed to confirm this assertion for multidimensional flows at all speeds.

## 5 Summary

A conservative, nonconservative, and hybrid time-derivative preconditioning method has been described. The conservative PROE method, which was demonstrated by the present authors to perform well on single component flows at all speeds [3], has been shown to lack robustness when applied to multicomponent flows and generates nonphysical oscillations near sharp component interfaces. The nonconservative PSCM method remains robust when applied to multicomponent flows and is accurate for flows without shocks, but converges to nonphysical solutions when shocks are present. The hybrid method, which combines the PROE and PSCM methods, retains the positive features of each of the methods resulting in a robust and efficient method for multicomponent flows at all speeds. In Part II of this paper each of the methods is applied to two-dimensional steady and unsteady multicomponent and multiphase flows, and similar behavior is observed.

## References


[1] Kiris, C., Kwak, D., Chan, W., and Housman, J., 2008, "High Fidelity Simulations for Unsteady Flow Through Turbopumps and Flowliners," Comput. Fluids, **37**, pp. 536–546.

[2] Kiris, C., Chan, W., Kwak, D., and Housman, J., "Time-Accurate Computational Analysis of the Flame Trench," ICCFD5, Seoul, Korea, Jul. 7–11.

[3] J. Housman, C. Kiris, and M. Hafez, "Preconditioned Methods for Simulations of Low Speed Compressible Flows," Comput. Fluids (to be published).

[4] Merkle, C. L., and Choi, Y. H., 1985, "Computation of Low Speed Compressible Flows With Time-Marching Methods," Int. J. Numer. Methods Eng., **25**, pp. 292–311.

[5] Turkel, E., 1987, "Preconditioning Methods for Solving Incompressible and Low-Speed Compressible Equations," J. Comput. Phys., **72**, pp. 277–298.

[6] Karni, S., 1994, "Multicomponent Flow Calculations by a Consistent Primitive Algorithm," J. Comput. Phys., **112**, pp. 31–43.

[7] Abgrall, R., 1996, "How to Prevent Pressure Oscillations in Multicomponent Flows," J. Comput. Phys., **125**, pp. 150–160.

[8] Karni, S., 1996, "Hybrid Multifluid Algorithms," SIAM J. Sci. Comput. (USA), **17**, pp. 1019–1039.

[9] Jenny, P., Muller, B., and Thomann, H., 1997, "Correction of Conservative Euler Solvers for Gas Mixtures," J. Comput. Phys., **132**, pp. 91–107.

[10] Abgrall, R., and Karni, S., 2001, "Computations of Compressible Multifluids," J. Comput. Phys., **169**, pp. 594–623.

[11] Chakravarthy, S. R., Anderson, D. A., and Salas, M. D., 1980, "The Split-Coefficient Matrix Method for Hyperbolic Systems of Gas Dynamics," 18th AIAA Aerospace Sciences Meeting, Paper No. AIAA-80-0268.

[12] Wren, G. P., Ray, S. E., Aliabadi, S. K., and Tezduyar, T. E., 1997, "Simulation of Flow Problems With Moving Mechanical Components, Fluid-Structure Interactions and Two-Fluid Interfaces," Int. J. Numer. Methods Fluids, **24**, pp. 1433–1448.

[13] Chorin, A. J., 1967, "A Numerical Method for Solving Incompressible Viscous Flow Problems," J. Comput. Phys., **2**, pp. 12–26.

[14] Kwak, D., Chang, J., Shanks, S., and Chakravarthy, S. R., 1986, "A Three-Dimensional Incompressible Navier-Stokes Solver Using Primitive Variables," AIAA J., **24**, pp. 390–396.

[15] Merkle, C. L., and Athavale, M., 1987, "Time-Accurate Unsteady Incompressible Flow Algorithms Based on Artificial Compressibility," AIAA Paper No. 87-1137.

[16] Rogers, S. E., Kwak, D., and Kiris, C., 1989, "Numerical Solution of the Incompressible Navier-Stokes Equations for Steady-State and Time-Dependent Problems," 27th AIAA Aerospace Sciences Meeting, Reno, NV, Paper No. AIAA-89-0463.

[17] Rehm, R. G., and Baum, H. R., 1978, "The Equations of Motion for Thermally Driven Buoyant Flows," J. Res. Natl. Bur. Stand., **83**, pp. 297–308.

[18] Klainerman, S., and Majda, A., 1981, "Singular Limits of Quasilinear Hyperbolic Systems With Large Parameters and the Incompressible Limit of Compressible Fluids," Commun. Pure Appl. Math., **34**, pp. 481–524.

[19] Klainerman, S., and Majda, A., 1982, "Compressible and Incompressible Fluids," Commun. Pure Appl. Math., **35**, pp. 629–653.

[20] Briley, W. R., McDonald, H., and Shamroth, S. J., 1983, "A Low Mach Number Euler Formulation and Application to Time-Iterative LBI Schemes," AIAA J., **21**, pp. 1467–1469.

[21] Viviand, H., 1985, "Pseudo-Unsteady Systems for Steady Inviscid Calculations," *Numerical Methods for the Euler Equations of Fluid Dynamics*, F. Angrand et al., eds., SIAM, Philadelphia, PA.

[22] Guerra, J., and Gustafsson, B., 1986, "A Numerical Method for Incompressible and Compressible Flow Problems With Smooth Solutions," J. Comput. Phys., **63**, pp. 377–397.

[23] Lindau, J. W., Kunz, R. F., Venkateswaran, S., and Merkle, C. L., 2001, "Development of a Fully-Compressible Multi-Phase Reynolds-Averaged Navier-Stokes Model," 15th AIAA Computational Fluid Dynamics Conference, Anaheim, CA, Paper No. AIAA-2001-2648.

[24] Edwards, J. R., 2001, "Toward Unified CFD Simulation of Real Fluid Flows," 15th AIAA Computational Fluid Dynamics Conference, Anaheim, CA, Paper No. AIAA-2001-2524.

[25] Li, D., Venkateswaran, S., Lindau, J. W., and Merkle, C. L., 2005, "A Unified Computational Formulation for Multi-Component and Multi-Phase Flows," 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, Paper No. AIAA-2005-1391.

[26] Edwards, J. R., and Liou, M.-S., 2006, "Simulation of Two-Phase Flows Using Low-Diffusion Shock-Capturing Schemes," 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, Paper No. AIAA-2006-1288.

[27] Neaves, M. D., and Edwards, J. R., 2006, "All-Speed Time-Accurate Underwater Projectile Caculations Using a Precoditioned Algorithm," ASME J. Fluids Eng., **128**, pp. 284–296.

[28] McDaniel, K. S., Edwards, J. R., and Neaves, M. D., 2006, "Simulation of Projectile Penetration Into Water and Sand," 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, Paper No. AIAA-2006-1289.

[29] Housman, J., 2007, "Time-Derivative Preconditioning Method for Multicomponent Flow," Ph.D. thesis, University of California Davis, Davis, CA.

[30] Lax, P. D., and Wendroff, B., 1960, "Systems of Conservation Laws," Commun. Pure Appl. Math., **13**, pp. 217–237.

[31] Moretti, G., 1979, "The Lambda-Scheme," Comput. Fluids, **7**, pp. 191–205.

[32] van Leer, B., Lee, W. T., and Roe, P. L., 1991, "Characteristic Time-Stepping or Local Preconditioning of the Euler Equations," AIAA Computational Fluid Dynamics Conference, Honolulu, HI, Paper No. AIAA-91-1552-CP.

[33] Roe, P. L., 1981, "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," J. Comput. Phys., **43**, pp. 357–372.

[34] Lombard, C. K., Bardina, J., Vankatapathy, E., and Oliger, J., 1983, "Multi-Dimensional Formulation of CSCM—An Upwind Flux Difference Eigenvector Split Method for the Compressible Navier-Stokes Equations," Sixth AIAA Computational Fluid Dynamics Conference, Danvers, MA, Paper No. AIAA-83-1895.

[35] Daywitt, J. E., Szostowski, D. J., and Anderson, D. A., 1983, "A Split-Coefficient/Locally Monotonic Scheme for Multishocked Supersonic Flow," AIAA J., **21**, pp. 871–880.

[36] Harabetian, E., and Pego, R., 1993, "Nonconservative Hybrid Shock Capturing Schemes," J. Comput. Phys., **105**, pp. 1–13.

[37] Toro, E. F., 1995, "On Adaptive Primitive-Conservative Schemes for Conservation Laws," *Sixth International Symposium on Computational Fluid Dynamics: A Collection of Technical Papers*, Vol. 3, M. M. Hafez, ed., pp. 1288–1293.

[38] Quirk, J. J., 1993, "Godunov-Type Schemes Applied to Detonation Flows," NASA/ICASE Contractor Report No. 191447.

[39] Ivings, M. J., Causon, D. M., and Toro, E. F., 1997, "On Hybrid High Resolution Upwind Methods for Multicomponent Flows," Z. Angew. Math. Mech., **77**, pp. 645–668.

[40] Chang, C.-H., and Liou, M.-S., 2005, "A Conservative Compressible Multifluid Model for Multiphase Flow: Shock-Interface Interaction Problems," 17th AIAA Computational Fluid Dynamics Conference, Paper No. AIAA 2005-5344.

**A. Corsini**
e-mail: alessandro.corsini@uniroma1.it

**F. Menichini**
e-mail: menichini@dma.ing.uniroma1.it

**F. Rispoli**
e-mail: franco.rispoli@uniroma1.it

Department of Mechanics and Aeronautics,
University of Rome "La Sapienza,"
Via Eudossiana 18,
Rome I00184, Italy

**A. Santoriello**
GE Oil and Gas,
Via Felice Matteucci 2,
Firenze 50127, Italy

**T. E. Tezduyar**
Mechanical Engineering,
Rice University,
MS 321 6100 Main Street,
Houston, TX 77005
e-mail: tezduyar@rice.edu

# A Multiscale Finite Element Formulation With Discontinuity Capturing for Turbulence Models With Dominant Reactionlike Terms

*A stabilization technique targeting the Reynolds-averaged Navier–Stokes (RANS) equations is proposed to account for the multiscale nature of turbulence and high solution gradients. The objective is effective stabilization in computations with the advection-diffusion reaction equations, which are typical of the class of turbulence scale-determining equations where reaction-dominated effects strongly influence the boundary layer prediction in the presence of nonequilibrium phenomena. The stabilization technique, which is based on a variational multiscale method, includes a discontinuity-capturing term designed to be operative when the solution gradients are high and the reactionlike terms are dominant. As test problems, we use a 2D model problem and 3D flow computation for a linear compressor cascade.* [DOI: 10.1115/1.3062967]

## 1 Introduction

Special-purpose computational fluid mechanics techniques targeting turbomachinery are becoming more and more effective in better understanding of the flow problems in this important application area. Many challenges, including the turbulent flow features, are still in need of improved modeling techniques. Examples of the efforts in this direction include stabilization methods for turbulence closures [1–3] and large eddy simulation (LES) techniques based on variational multiscale methods [4–6].

The physics of turbulent flows in turbomachinery configurations is governed by nonequilibrium phenomena that cannot be addressed adequately with the Boussinesq effective viscosity concept. This is because of the presence of the curvature and rotation effects and large separation and recirculation regions. Even when tackled with advanced turbulence closures such as nonisotropic first order models or Reynolds-stress models, flow simulations involve numerical shortcomings that are not fully addressed by standard stabilization methods developed in the context of advection-diffusion type equations.

The numerical counterpart of the nonequilibrium phenomena is that the flow is governed by scale-determining equations with dominant reactionlike terms, stemming from turbulence dissipation mechanisms involved. For example, reactionlike terms become dominant in stagnation regions, separated boundary layers, and recirculating flow cores, where the flow velocity approaches zero.

In recent decades, a number of studies focused on stabilized formulations for advection-diffusion reaction equations. These include equations governing chemically reacting flows and equations with numerically generated reactionlike terms. As examples of such studies, we can mention the diffusion for reaction dominated (DRD) method by Tezduyar and Park [7,8], studies by Codina [9], unusual stabilized finite element method (USFEM) by Franca and Valentin [10], SPG by Corsini et al. [1], and stabilized methods emanating from the variational multiscale (VMS) concept [11], such as the ones described in Refs. [12,2,13].

In this paper, we describe a stabilization technique targeting the Reynolds-averaged Navier–Stokes (RANS) equations, accounting for the multiscale nature of turbulence and the high solution gradients involved. The technique is based on the variable sub grid scale (VSGS) formulation [2] and includes discontinuity capturing in the form of a new generation DRD method [3]. The objective in the approach we take here is to accomplish the additional stabilization without affecting the accuracy in advection-dominated zones and in zones where the solution is smooth. The main application area we have in mind is turbomachinery. We are focusing on addressing the numerical challenges posed by the reactionlike terms appearing in the closure equations for advanced eddy viscosity models, such as the nonlinear $k$-$\varepsilon$ model [14].

In Sec. 2, we provide an overview of the nonlinear $k$-$\varepsilon$ model and the strong formulation of the RANS problem. The variational multiscale formulation for the RANS equations is described in Sec. 3. In Sec. 4, we describe the stabilization parameters, discontinuity capturing, and the DRD method, including the DRDJ method, which takes into account the local "jump" in the solution. The model test problem is presented in Sec. 5 and the 3D flow computation of a linear compressor cascade in Sec. 6. Concluding remarks are given in Sec. 7.

## 2 RANS Formulation for Incompressible Turbulent Flows

Let $\Omega \subset \mathbb{R}^{nsd}$ be the spatial domain with boundary $\Gamma$ and $(0,T)$ be the time domain. The unsteady RANS equations of incompressible flows can be written on $\Omega$ and $\forall t \in (0,T)$ as

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} - \mathfrak{I}\right) - \nabla\cdot\boldsymbol{\sigma} = 0 \tag{1}$$

Contributed by the Applied Mechanics Division of ASME for publication in the JOURNAL OF APPLIED MECHANICS. Manuscript received May 29, 2008; final manuscript received August 18, 2008; published online February 5, 2009. Review conducted by Arif Masud.

**Table 1 Closure coefficients in $k$ and $\varepsilon$ equations [13]**

| | |
|---|---|
| $c_\mu$ | $\dfrac{0.3[1-\exp[-0.36/\exp[-0.75\max(\hat{e},\hat{\varpi})]]]}{1+0.35[\max(\hat{e},\hat{\varpi})]^{1.5}}$ |
| $f_\mu$ | $1-\exp[-(\mathrm{Re}_t/90)^{0.5}-(\mathrm{Re}_t/400)^2]$ |
| $c_{\varepsilon 1}$ | 1.44 |
| $c_{\varepsilon 2}$ | 1.92 |
| $f_{\varepsilon 2}$ | $[1-0.3\exp(-\mathrm{Re}_t^2)]$ |
| $\sigma_\varepsilon$ | 1.3 |
| $\sigma_k$ | 1 |

$$\nabla\cdot\boldsymbol{u}=0 \tag{2}$$

$$\rho\left(\frac{\partial\boldsymbol{\phi}}{\partial t}+\boldsymbol{u}\cdot\nabla\boldsymbol{\phi}+\boldsymbol{B}_{k\varepsilon}\boldsymbol{\phi}-\mathfrak{I}_{k\varepsilon}\right)-\nabla\cdot(\rho\boldsymbol{\nu}_{k\varepsilon}(\nabla\boldsymbol{\phi}))=0 \tag{3}$$

where $\rho$ is the density, $\boldsymbol{u}$ is the velocity vector, $\boldsymbol{\phi}=(k,\tilde{\varepsilon})^T$, and $k$ and $\tilde{\varepsilon}$ are the turbulent kinetic energy and the homogeneous dissipation variable. The symbols $\mathfrak{I}$ and $\mathfrak{I}_{k\varepsilon}$ represent the vector of external forces and the source vector of turbulent scale-determining equations. As proposed in Ref. [15], $\mathfrak{I}$ accounts for the volume sources related to the second and third-order terms in the nonisotropic stress-strain constitutive relation [14]. The force vector reads as

$$\begin{aligned}\mathfrak{I}=\nabla\cdot\Big[&-0.1\nu_t\tau\left(\boldsymbol{\varepsilon}(\boldsymbol{u})\cdot\boldsymbol{\varepsilon}(\boldsymbol{u})-\boldsymbol{\varepsilon}(\boldsymbol{u}):\boldsymbol{\varepsilon}(\boldsymbol{u})\tfrac{1}{3}\boldsymbol{I}\right)\\&+0.1\nu_t\tau(\boldsymbol{\varpi}(\boldsymbol{u})\cdot\boldsymbol{\varepsilon}(\boldsymbol{u})+(\boldsymbol{\varpi}(\boldsymbol{u})\cdot\boldsymbol{\varepsilon}(\boldsymbol{u}))^{\mathrm{T}})\\&+0.26\nu_t\tau\left(\boldsymbol{\varpi}(\boldsymbol{u})\cdot\boldsymbol{\varpi}(\boldsymbol{u})-\boldsymbol{\varpi}(\boldsymbol{u}):\boldsymbol{\varpi}(\boldsymbol{u})\tfrac{1}{3}\boldsymbol{I}\right)\\&-10c_\mu^2\nu_t\tau^2(\boldsymbol{\varepsilon}(\boldsymbol{u})\cdot\boldsymbol{\varepsilon}(\boldsymbol{u})\cdot\boldsymbol{\varpi}(\boldsymbol{u})+(\boldsymbol{\varepsilon}(\boldsymbol{u})\cdot\boldsymbol{\varepsilon}(\boldsymbol{u})\cdot\boldsymbol{\varpi}(\boldsymbol{u}))^T)\\&-5c_\mu^2\nu_t\tau^2(\boldsymbol{\varepsilon}(\boldsymbol{u}):\boldsymbol{\varepsilon}(\boldsymbol{u}))\boldsymbol{\varepsilon}(\boldsymbol{u})+5c_\mu^2\nu_t\tau^2(\boldsymbol{\varpi}(\boldsymbol{u}):\boldsymbol{\varpi}(\boldsymbol{u}))\boldsymbol{\varepsilon}(\boldsymbol{u})\Big]\end{aligned} \tag{4a}$$

Here, $\boldsymbol{\varepsilon}(\boldsymbol{u})=((\nabla\boldsymbol{u})+(\nabla\boldsymbol{u})^T)$ is twice the strain-rate tensor, $\boldsymbol{\varpi}(\boldsymbol{u})=((\nabla\boldsymbol{u})-(\nabla\boldsymbol{u})^T)$ is twice the vorticity tensor, $\nu_t$ is the turbulent kinematic viscosity defined as $\nu_t=c_\mu f_\mu\tau k$, and $\tau=k/\bar{\varepsilon}$ is the turbulence time scale, with $c_\mu$ and $f_\mu$ and other closure coefficients for the turbulence model [14] used given in Table 1.

In Table 1, $\mathrm{Re}_t=k^2/(\nu\bar{\varepsilon})$ is the turbulence Reynolds number, and $\hat{e}$ and $\hat{\varpi}$ are, respectively, the strain-rate and vorticity invariants defined as $\hat{e}=\tau\sqrt{0.5\boldsymbol{\varepsilon}(\boldsymbol{u}):\boldsymbol{\varepsilon}(\boldsymbol{u})}$ and $\hat{\varpi}=\tau\sqrt{0.5\boldsymbol{\varpi}(\boldsymbol{u}):\boldsymbol{\varpi}(\boldsymbol{u})}$.

The source vector $\mathfrak{I}_{k\varepsilon}$ is defined as

$$\mathfrak{I}_{k\varepsilon}=\begin{bmatrix}P_k-\rho D\\ c_{\varepsilon 1}P_k\dfrac{\tilde{\varepsilon}}{k}+E\end{bmatrix} \tag{4b}$$

where $P_k=\rho\boldsymbol{R}:\nabla\boldsymbol{u}$ is the production of turbulent kinetic energy with $\boldsymbol{R}$ the Reynolds-stress tensor, $D=2\nu\nabla\sqrt{k}\cdot\nabla\sqrt{k}$ is the dissipation rate value on solid boundaries, and $E=0.0022\hat{e}k\tau\nu_t\|\nabla\cdot(\nabla\boldsymbol{u})\|^2$ is the near-wall additional source.

The stress tensor, in the momentum equation, is defined as

$$\boldsymbol{\sigma}(p,\boldsymbol{u})=-\left(p+\frac{2}{3}k\right)\boldsymbol{I}+\rho\nu_{\boldsymbol{u}}\boldsymbol{\varepsilon}(\boldsymbol{u}) \tag{5}$$

with $\nu_{\boldsymbol{u}}=\nu+\nu_t$, where $\nu$ is the molecular viscosity.

The diffusion terms in the turbulent scale-determining equations depend on a diffusivity matrix defined as

$$\boldsymbol{\nu}_{k\varepsilon}=\begin{bmatrix}\left(\nu+\dfrac{\nu_t}{\sigma_k}\right) & 0\\ 0 & \left(\nu+\dfrac{\nu_t}{\sigma_\varepsilon}\right)\end{bmatrix} \tag{6}$$

with the value of the coefficients $\sigma_k$ and $\sigma_\varepsilon$ given in Table 1.

The reaction terms, absorption like in Eq. (3), account for the dissipation-destruction matrices and are defined as

$$\boldsymbol{B}_{k\varepsilon}\boldsymbol{\phi}=\begin{bmatrix}B_k & 0\\ 0 & B_{\varepsilon}\end{bmatrix}\cdot\boldsymbol{\phi} \tag{7a}$$

with

$$B_k=\frac{\tilde{\varepsilon}}{k},\quad B_\varepsilon=c_{\varepsilon 2}f_{\varepsilon 2}\frac{\tilde{\varepsilon}}{k} \tag{7b}$$

The essential and natural boundary conditions for Eqs. (1) and (3) are represented as

$$\boldsymbol{u}=\boldsymbol{g}\quad\text{and}\quad\boldsymbol{\phi}=\boldsymbol{g}_{k\varepsilon}\quad\text{on}\quad\Gamma_g \tag{8a}$$

$$\boldsymbol{n}.\boldsymbol{\sigma}=\boldsymbol{h}\quad\text{and}\quad\boldsymbol{n}\cdot(\rho\boldsymbol{\nu}_{k\varepsilon}(\nabla\boldsymbol{\phi}))=0\quad\text{on}\quad\Gamma_h \tag{8b}$$

where $\Gamma_g$ and $\Gamma_h$ are the complementary subsets of the boundary $\Gamma$, $\boldsymbol{n}$ is the direction normal to the boundary, and $\boldsymbol{g}$, $\boldsymbol{g}_{k\varepsilon}$, and $\boldsymbol{h}$ are the given functions representing the essential and natural boundary conditions.

## 3 Variational Multiscale Formulation for RANS Equations

In describing the VSGS formulation of Eqs. (1) and (3), we assume that we have constructed some suitably defined finite-dimensional trial solution and test function spaces $\mathrm{S}_{\boldsymbol{u}}^h,\mathrm{S}_p^h,\mathrm{S}_{\boldsymbol{\phi}}^h$ and $\mathbb{V}_{\boldsymbol{u}}^h,\mathbb{V}_p^h,\mathbb{V}_{\boldsymbol{\phi}}^h$.

The variational multiscale formulation, based on the VSGS method [2], reads as

find $\boldsymbol{u}^h\in\mathrm{S}_{\boldsymbol{u}}^h,p^h\in\mathrm{S}_p^h,\boldsymbol{\phi}^h\in\mathrm{S}_{\boldsymbol{\phi}}^h$ such that $\forall\,\boldsymbol{w}^h\in\mathbb{V}_{\boldsymbol{u}}^h$, $\forall\,q^h\in\mathbb{V}_p^h$,

and $\forall\,\boldsymbol{\psi}^h\in\mathbb{V}_{\boldsymbol{\phi}}^h$

$$\begin{aligned}&\int_\Omega\boldsymbol{w}^h\cdot\left(\frac{\partial\boldsymbol{u}^h}{\partial t}+\boldsymbol{u}^h\cdot\nabla\boldsymbol{u}^h-\mathfrak{I}^h\right)d\Omega+\int_\Omega\boldsymbol{\varepsilon}(\boldsymbol{w}^h):\boldsymbol{\sigma}(p^h,\boldsymbol{u}^h)d\Omega\\&\quad-\int_{\Gamma^h}\boldsymbol{w}^h\cdot\boldsymbol{h}^h d\Gamma+\int_\Omega q^h\nabla\cdot\boldsymbol{u}^h d\Omega\\&\quad+\sum_{e=1}^{n_{\mathrm{el}}}\int_{\Omega^e}\boldsymbol{P}^{\mathrm{stab}}(\boldsymbol{w}^h,q^h)\cdot[\mathcal{L}(p^h,\boldsymbol{u}^h)-\rho\mathfrak{I}^h]d\Omega\\&\quad+\sum_{e=1}^{n_{\mathrm{el}}}\int_{\Omega^e}\nu_{\mathrm{DCDD}}\rho\nabla\boldsymbol{w}^h:\nabla\boldsymbol{u}^h d\Omega=0\end{aligned} \tag{9a}$$

where

$$\mathcal{L}(q^h,\boldsymbol{w}^h)=\rho\left(\frac{\partial\boldsymbol{w}^h}{\partial t}+\boldsymbol{u}^h\cdot\nabla\boldsymbol{w}^h\right)-\nabla\cdot\boldsymbol{\sigma}(q^h,\boldsymbol{w}^h) \tag{9b}$$

and

$$\int_\Omega \boldsymbol{\psi}^h \cdot \rho\left(\frac{\partial \boldsymbol{\phi}^h}{\partial t} + \boldsymbol{u}^h \cdot \nabla \boldsymbol{\phi}^h + \boldsymbol{B}_{k\varepsilon}\boldsymbol{\phi} - \mathfrak{I}^h_{k\varepsilon}\right) d\Omega$$

$$+ \int_\Omega \nabla \boldsymbol{\psi}^h \cdot (\rho \boldsymbol{\nu}_{k\varepsilon}(\nabla \boldsymbol{\phi}^h)) d\Omega - \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \boldsymbol{P}^{\text{stab}}_{k\varepsilon}(\boldsymbol{\psi}^h) \cdot [\text{Ł}_{k\varepsilon}(\boldsymbol{\phi}^h)$$

$$- \rho \mathfrak{I}^h_{k\varepsilon}] d\Omega + \sum_{e=1}^{n_{\text{el}}} \int_{\Omega^e} \mathbf{K}^{\text{DC}}_{k\varepsilon} \rho \nabla \boldsymbol{\psi}^h : \nabla \boldsymbol{\phi}^h d\Omega = 0 \tag{10a}$$

where

$$\text{Ł}_{k\varepsilon}(\boldsymbol{\phi}^h) = \rho\left(\frac{\partial \boldsymbol{\phi}^h}{\partial t} + \boldsymbol{u}^h \cdot \nabla \boldsymbol{\phi}^h + \boldsymbol{B}_{k\varepsilon}\boldsymbol{\phi}^h\right) - \nabla \cdot (\rho \boldsymbol{\nu}_{k\varepsilon}(\nabla \boldsymbol{\phi}^h)) \tag{10b}$$

Here $\boldsymbol{P}^{\text{stab}}$, $\boldsymbol{P}^{\text{stab}}_{k\varepsilon}$, and $\mathbf{K}^{\text{DC}}_{k\varepsilon}$ are, respectively, the VSGS stabilization operators and the dissipation matrix for the discontinuity-capturing (DC) scheme, while $\nu_{\text{DCDD}}$ is for the discontinuity-capturing directional dissipation (DCDD) stabilization [6,16]. The definition of $\nu_{\text{DCDD}}$ is given in Sec. 4.

A fundamental result [2] is the demonstration that VSGS stabilization can be seen as a particular form of the Petrov–Galerkin operator, thus the vectors $\boldsymbol{P}^{\text{stab}}$ and $\boldsymbol{P}^{\text{stab}}_{k\varepsilon}$ take the following forms:

$$\boldsymbol{P}^{\text{stab}}(\boldsymbol{w}^h) = \tau_{\text{VSGS}} \rho (\boldsymbol{u}^h \cdot \nabla) \boldsymbol{w}^h + \tau_{\text{PSPG}} \nabla q \tag{11a}$$

$$\boldsymbol{P}^{\text{stab}}_{k\varepsilon}(\boldsymbol{\psi}^h) = \begin{bmatrix} \tau_{\text{VSGS-}k} & 0 \\ 0 & \tau_{\text{VSGS-}\varepsilon} \end{bmatrix} \rho (\boldsymbol{u}^h \cdot \nabla) \boldsymbol{\psi}^h \tag{11b}$$

with $\tau_{\text{VSGS}}$ and $\tau_{\text{PSPG}}$ being the VSGS and PSPG stabilization parameters, the latter for equal-order velocity-pressure approximations. In Eqs. (9*a*) and (10*a*), the VSGS parameters are defined as the product of elementwise variable intrinsic time scale $\tau_{\text{VSGS}}$. These are defined in Sec. 4.

The dissipation terms for advection-diffusion reaction equations are defined as

$$\mathbf{K}^{\text{DC}}_{k\varepsilon} = \begin{bmatrix} \kappa_{\text{DRDJ-}k} & 0 \\ 0 & \kappa_{\text{DRDJ-}\varepsilon} \end{bmatrix} \tag{12}$$

Here $\kappa_{\text{DRDJ-}k}$ and $\kappa_{\text{DRDJ-}\varepsilon}$ are the DRDJ additional diffusivities, also defined in Sec. 4.

## 4 Stabilization Parameters and Discontinuity Capturing

**4.1 VSGS Parameters.** The intrinsic time scale $\tau_{\text{VSGS}}$, which provides the subgrid scale residual modeling as proposed in Ref. [2], is derived by the combination of one-dimensional intrinsic time scale parameters, associated with each parent-domain coordinate direction [2].

For $\boldsymbol{u}$ and $\boldsymbol{\phi}=(k,\tilde{\varepsilon})^T$, the time scales are defined along each parent coordinate as the product of elementwise time scale $\tau_{\text{SC}_{\xi_i}}$ and the space-dependent function $f_{\xi i}$ [2], on the basis of directional Péclet numbers

$$(\text{Pe}_{\xi i})_{\boldsymbol{u}} = \frac{|u_{\xi i}| h_{\text{UGN}}}{2\nu_{\boldsymbol{u}}} \tag{13a}$$

$$(\text{Pe}_{\xi i})_k = \frac{|u_{\xi i}| h_{\text{UGN}}}{2\left(\nu + \dfrac{\nu_t}{\sigma_k}\right)} \tag{13b}$$

$$(\text{Pe}_{\xi i})_\varepsilon = \frac{|u_{\xi i}| h_{\text{UGN}}}{2\left(\nu + \dfrac{\nu_t}{\sigma_\varepsilon}\right)} \tag{13c}$$

and reaction numbers

$$\beta^2_k = \frac{B_k}{\left(\nu + \dfrac{\nu_t}{\sigma_k}\right)} \frac{h^2_{\text{RGN-}k}}{4}, \beta^2_\varepsilon = \frac{B_\varepsilon}{\left(\nu + \dfrac{\nu_t}{\sigma_\varepsilon}\right)} \frac{h^2_{\text{RGN-}\varepsilon}}{4} \tag{13d}$$

In the above definitions, the element length in the advection-dominated limit [7] is

$$h_{\text{UGN}} = 2\left(\sum_a |\boldsymbol{s} \cdot \nabla N_a|\right)^{-1} \tag{14a}$$

where $\boldsymbol{s}$ is the unit vector in direction of the velocity.

In the diffusion-reaction dominated limit, the element length turns to the corresponding scale [16,17]:

$$h_{\text{RGN-}k} = 2\left(\sum_a |\boldsymbol{r}_k \cdot \nabla N_a|\right)^{-1} \tag{14b}$$

$$h_{\text{RGN-}\varepsilon} = 2\left(\sum_a |\boldsymbol{r}_\varepsilon \cdot \nabla N_a|\right)^{-1} \tag{14c}$$

where $\boldsymbol{r}_k$ and $\boldsymbol{r}_\varepsilon$ are the unit vectors in the direction of the solution gradient defined as

$$\boldsymbol{r}_k = \frac{\nabla |k|}{\|\nabla |k|\|}, \quad \boldsymbol{r}_\varepsilon = \frac{\nabla |\tilde{\varepsilon}|}{\|\nabla |\tilde{\varepsilon}|\|} \tag{14d}$$

*Remark 1*. The reactionlike terms, not considered in most of the stabilized formulations found in the literature, could be important in turbulence computation. When considering the magnitude of the reaction-to-advection ratios, reaction driven phenomena affect the flow in the near-wall region and are emphasized in nonequilibrium phenomena such as in the stagnation, transition, or separation regions. Corsini et al. [3] recently demonstrated that approaching a solid wall, the reaction-to-advection ratio behaves like $1/d^2_w$, with $d_w$ being the distance from the solid wall.

The multidimensional time-scale parameter, in its space-time version, is computed by using the *r*-switch [18]:

$$\tau_{\text{VSGS}} = \left(\frac{1}{\left(\dfrac{1}{\tau^{\text{space}}_{\text{VSGS}}}\right)^{r_s} + \left(\dfrac{1}{\tau^{\text{time}}_{\text{VSGS}}}\right)^{r_s}}\right)^{1/r_s} \tag{15}$$

where

$$\tau^{\text{space}}_{\text{VSGS}} = \left(\frac{1}{\left(\dfrac{1}{\tau_{\text{SC}_\xi}}\right)^{r_s} + \left(\dfrac{1}{\tau_{\text{SC}_\eta}}\right)^{r_s} + \left(\dfrac{1}{\tau_{\text{SC}_\eta}}\right)^{r_s}}\right)^{1/r_s}$$

$$(1 + f_\xi(\xi, \text{Pe}_\xi, \beta^2)) \cdot (1 + f_\eta(\eta, \text{Pe}_\eta, \beta^2)) \cdot (1 + f_\zeta(\zeta, \text{Pe}_\zeta, \beta^2)) \tag{16a}$$

with $\beta=0$ for the flow equations, and

$$\tau^{\text{time}}_{\text{VSGS}} = \frac{\Delta t}{2} \tag{16b}$$

**4.2 Discontinuity-Capturing Parameters.** The DCDD viscosity is defined by using the expression from [6,16]

$$\nu_{\text{DCDD}} = \tau_{\text{DCDD}} \|\boldsymbol{u}^h\|^2 \tag{17a}$$

with

$$\tau_{\text{DCDD}} = \frac{h_{\text{DCDD}}}{2u_{\text{ref}}} \frac{\|\nabla \|\boldsymbol{u}^h\|\| h_{\text{DCDD}}}{u_{\text{ref}}} \tag{17b}$$

Here $h_{\text{DCDD}}$ is the element length scale in the diffusion dominated limit [16,17], which reads as

$$h_{\text{DCDD}} = h_{\text{RGN}} = 2\left(\sum_a |\boldsymbol{r} \cdot \nabla N_a|\right)^{-1} \tag{18a}$$

and $\boldsymbol{r}$ is the unit vector in the direction of the solution gradient, defined as

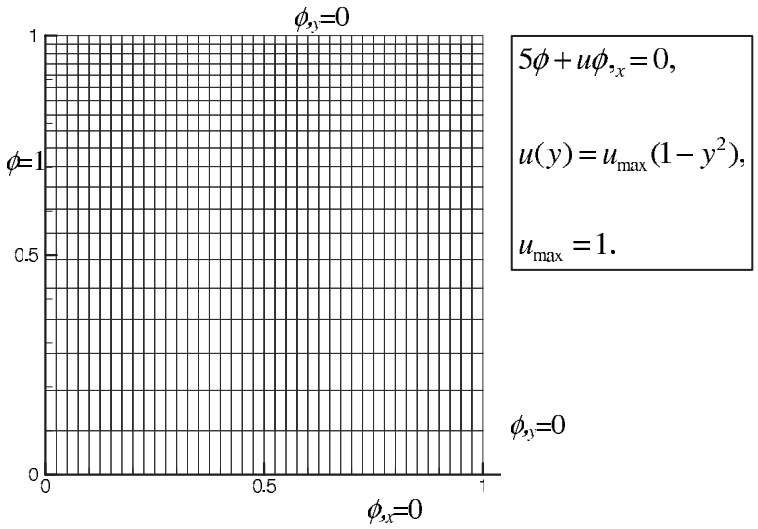



**Fig. 1 Scalar test case. Problem statement, and grid and boundary conditions.**

$$r = \frac{\nabla\|\boldsymbol{u}\|}{\|\nabla\|\boldsymbol{u}\|\|} \tag{18b}$$

As far as the reaction-dominated limit is concerned, the original DRD method from Tezduyar and Park [7] was proposed as a remedy for the numerical instabilities in Eq. (10*a*). The DRD method was obtained for two limit cases: advection-reaction and diffusion-reaction. For both cases, the analytical expression for the additional DRD term depends on dimensionless numbers that relate, respectively, the reaction rate to the advection and diffusion rates, taking into account the quality of the grid used.

Recently [3], a new stabilization technique named DRDJ was formulated, and this takes into account the local variation ("jump") in the solution, turning to a DRD-like discontinuity-capturing scheme.

The DRDJ additional diffusivity reads, for advection-reaction limit, as

$$\kappa_{\mathrm{AR}}(\gamma_\phi, J_e) = \frac{1}{2} u h_{\mathrm{UGN}} J_e \left( -\coth\ \gamma_\phi + \gamma_\phi \left( \frac{1}{\sinh^2\ \gamma_\phi} + 4\alpha \right) \right) \tag{19a}$$

while for the diffusion-reaction limit it is defined as

$$\kappa_{\mathrm{DR}}(\beta_\phi, J_e) = B_\phi \left( \frac{h_{\mathrm{RGN}\text{-}\phi}}{2} \right)^2 J_e \left( 4\alpha + \frac{1}{\sinh^2\ \beta_\phi} - \frac{1}{\beta_\phi^2} \right) \tag{19b}$$

where the subscript $\phi = k, \varepsilon$ generates the expressions corresponding to $k$ and $\varepsilon$, with $\gamma_k$ and $\gamma_\varepsilon$ defined as

$$\gamma_k = \frac{B_k}{\|\boldsymbol{u}\|} \frac{h_{\mathrm{UGN}}}{2} \tag{20a}$$

$$\gamma_\varepsilon = \frac{B_\varepsilon}{\|\boldsymbol{u}\|} \frac{h_{\mathrm{UGN}}}{2} \tag{20b}$$

Here, $J_e$ is a normalized measure of the variation (jump) in the solution over an element. In Eqs. (19*a*) and (19*b*), $\alpha$ is a parameter for the integration rule of the element reaction coefficient matrix (e.g. $\alpha$ equal to 1/6 for two-point Gaussian quadrature and $O$ for the "lumped" case).

The resulting DRDJ could be generalized to a multidimensional diffusivity tensor:

$$\boldsymbol{\kappa}_{\mathrm{DRDJ}\text{-}\phi} = \kappa_{\mathrm{AR}}(\gamma_\phi, J_e)\boldsymbol{ss} + \kappa_{\mathrm{DR}}(\beta_\phi, J_e)(\boldsymbol{tt} + \boldsymbol{vv}) \tag{21}$$

where $\boldsymbol{t}$ and $\boldsymbol{v}$ are the two unit vectors orthogonal to $\boldsymbol{s}$ and to each other. We note that along the $\boldsymbol{t}$ and $\boldsymbol{v}$ directions the numerical diffusion is the one associated with the one-dimensional diffusion-reaction case. For this reason, the element characteristic length measure is provided by the $h_{\mathrm{RGN}}$ [16,17].

*Remark 2.* The parameter $J_e$ is defined as follows:

$$J_e = \frac{(\phi_{\max})_e - (\phi_{\min})_e}{\|\phi\|_e} \tag{22}$$

where $(\phi_{\max})_e$ and $(\phi_{\min})_e$ are the maximum and minimum values of the variable $\phi$ for element $e$. Here $\|\phi\|_e$ represents a local scaling for the unknown, which could be set equal to a global scaling. Turbulent flow parameters are characterized by different orders of magnitude for different zones of the flow field, thus for turbulence computations $\|\phi\|_e$ in a way takes into account the local features of the problem and is chosen equal to the maximum value of the unknown in the element. With such a choice, as it is done for the calculations presented here, we assure that $J_e$ ranges from 0 to 1, thus leading to a diffusivity that is everywhere limited.

## 5 Scalar Test Case

The scalar test problem was first proposed in Ref. [7] as a test for the original DRD formulation. The square domain is discretized with of a nonisotropic Cartesian grid with $41 \times 21$ linear finite elements, and the problem statement is given in Fig. 1.

Figure 2 shows the solutions with the SUPG, SUPG+DRDJ, and VSGS+DRDJ. Figure 3 shows some solution profiles extracted from near the boundaries where the reaction dominates advection. The exact and SUPG plus mass lumping (SUPG +ML) solutions are also plotted to provide comparison.

The SUPG+DRDJ solution exhibits a 0.05 undershoot along the second row of nodes (see Figs. 2(*c*) and 3(*a*)). This local oscillation is eliminated in the SUPG+ML solution but with an overdiffusive layer, and in VSGS+DRDJ solution, which produces the sharpest undisturbed solution layer. For the third row of nodes (Fig. 3(*b*)), the DRDJ solutions are all very close to the exact solution but SUPG+ML still shows an overdiffusive behavior.

From these results, we conclude that stabilization methods designed to remedy instabilities due to dominant advection terms cannot control instabilities due to dominant reactionlike terms.

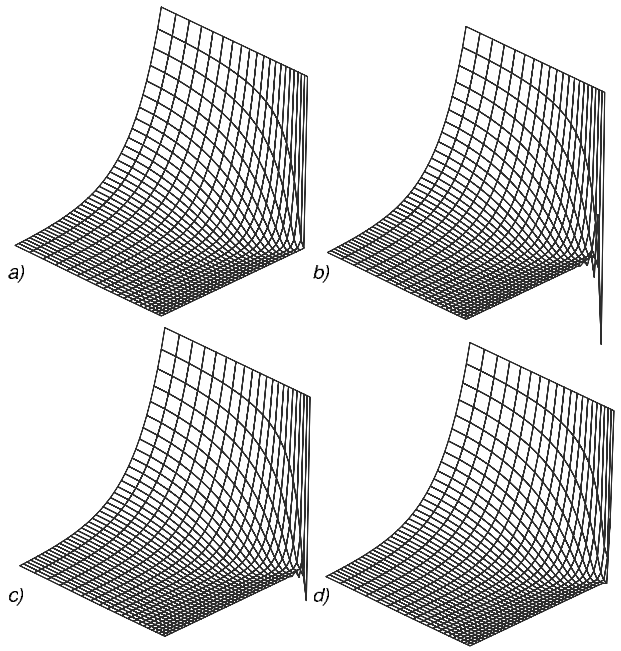



**Fig. 2 Scalar test case. (*a*) Exact solution, (*b*) SUPG, (*c*) SUPG+DRDJ, and (*d*) VSGS+DRDJ.**

Some additional stabilization, different from mass lumping, which is overdiffusive, is needed. The DRD formulation serves that purpose while retaining the solution accuracy.

## 6 Turbulent Flow Test Case: NACA65 Compressor Cascade

We considered the tip leakage flow in a linear compressor cascade of NACA65-1810 profile with flat ends. The main cascade design parameters are shown in Fig. 4 together with the computational grid. The cascade has been experimentally investigated by Kang and Hirsch [19,20]. The cascade has a tip clearance of 2% of the chord length, and the inlet flow angle is equal to 29.3 deg with respect to the streamwise direction.

The cascade flow is simulated in near-design incidence condition, with the flow regarded as incompressible and steady. In accordance with the experimental findings, the flow at the inlet is fully turbulent, i.e., the measured shape factor is about 1.22 at 40% of the chord length upstream of the leading edge. The Reynolds number is based on the chord length and the inlet bulk velocity is $3\times10^5$. The experimental freestream turbulence intensity is 3.4%. The dissipation length scale $l_\varepsilon$ is set equal to 5% of the chord length.

The solution domain encompasses the upper half of the blade span (i.e., 104 mm from the casing to midspan, expecting the flow to be symmetric in the spanwise direction), bounded by the two symmetry planes in the pitchwise direction, and stretches 40% of the chord length upstream from the blade leading edge and one chord downstream from the trailing edge.

The coordinate system used is orthogonal, with $x$, $y$, and $z$ denoting the streamwise, pitchwise, and spanwise directions, respectively. An embedded $H$-topology computational mesh with a total of about $0.8\times10^6$ nodes with $Q1$-$Q1$ elements is adopted. There are 141 nodes in the streamwise, 73 nodes in the pitchwise, and 81 nodes in the spanwise directions, respectively. The mesh is clustered around the blade walls, the leading and trailing edges, and in the wake. The first wall-adjacent $y+$ values were everywhere lower than 1.6. A uniform discretization (with 21 points) is used to resolve the tip gap.

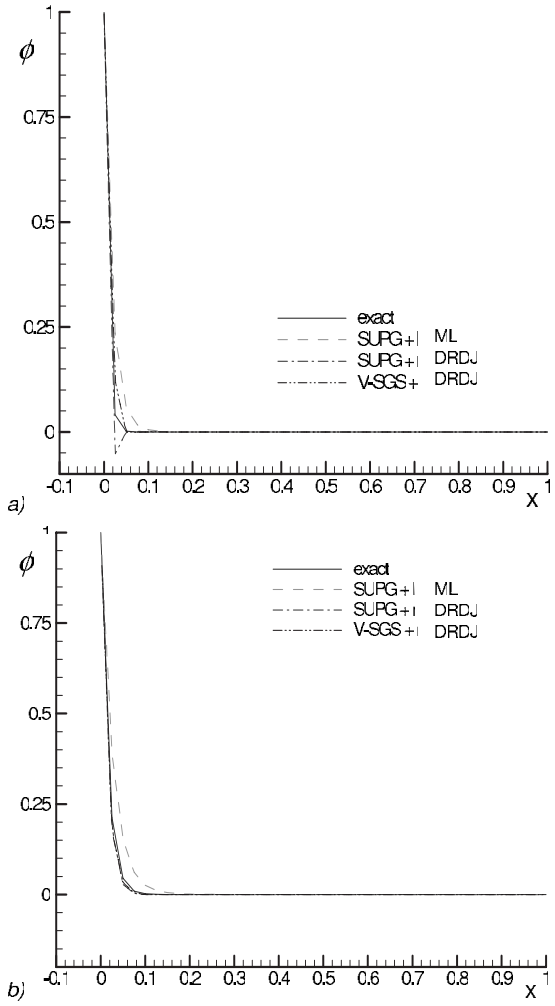
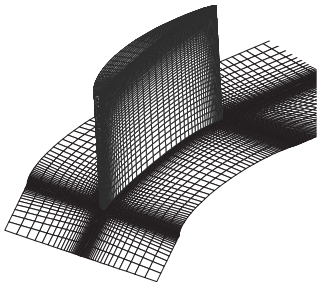



**Fig. 3 Scalar test case. Profiles extracted at (*a*) second and (*b*) third rows of nodes.**

The position of the inlet is regarded as being sufficiently distant to eliminate any upstream effect of the outflow conditions on the solution both in the cascade passage and in the wake region.

Figure 5 shows the static pressure coefficient $C_p$ computed by using SUPG+DRDJ and VSGS+DRDJ. The figure shows that the two numerical solutions exhibit similar behaviors, being both in qualitative agreement with the experimental results. When comparing the pressure fields in the region where the leakage flow develops and rolls-up, the VSGS+DRDJ provides a nicer and sharper representation of the isobar troughs tracing the vortex core path. This is due to the reaction limit control in the vortex defect region.



| *NACA65 cascade geometry* | |
|---|---|
| *profile family* | NACA65 – 1810 |
| *aspect ratio* | 1.0 |
| *chord* | 200 mm |
| *spacing* | 180 mm |
| *solidity* | 1.111 |
| *stagger angle* | 10° |

**Fig. 4 Cascade geometry and computational grid**

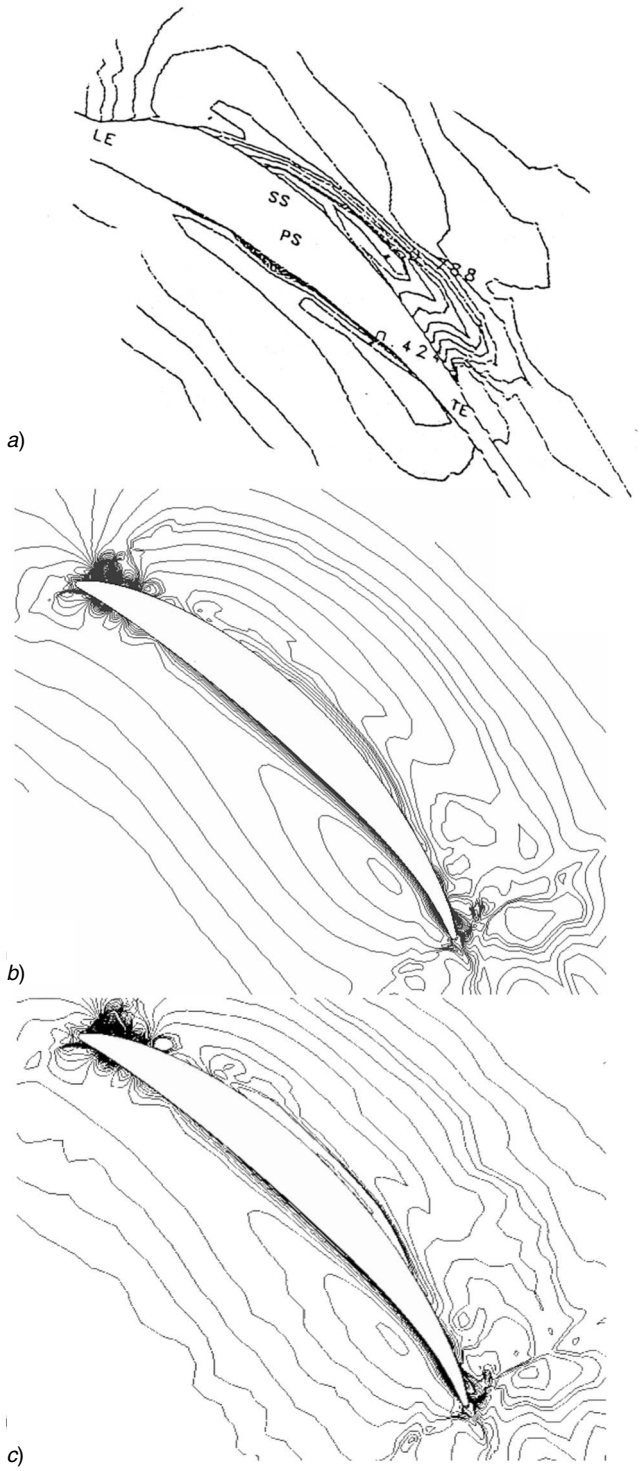



**Fig. 5 Static pressure coefficient isolines in the tip gap. (*a*) Experiments [19], (*b*) SUPG+DRDJ, and (*c*) VSGS+DRDJ.**

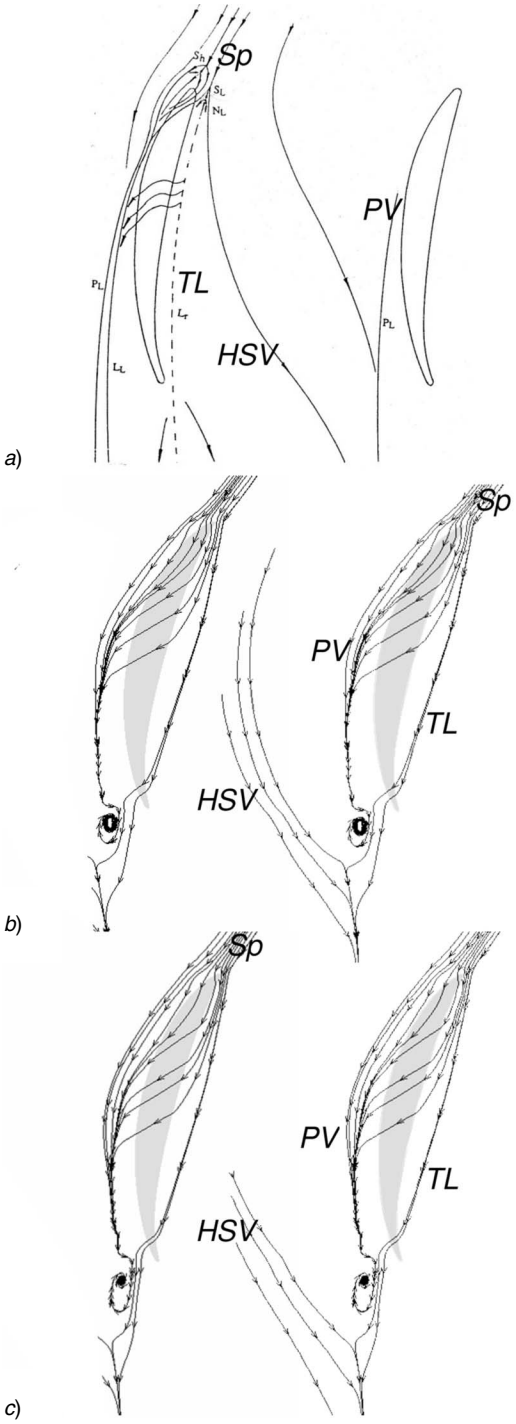
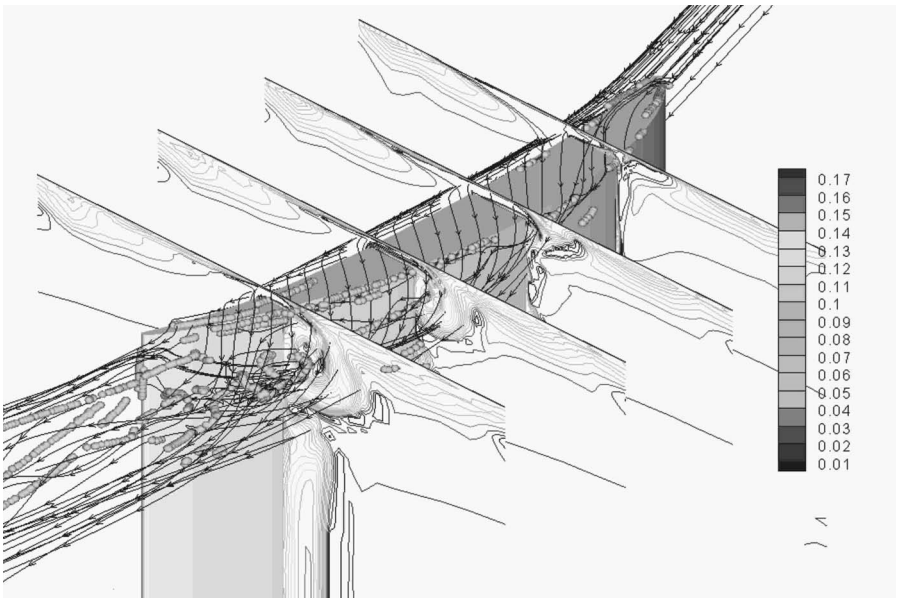


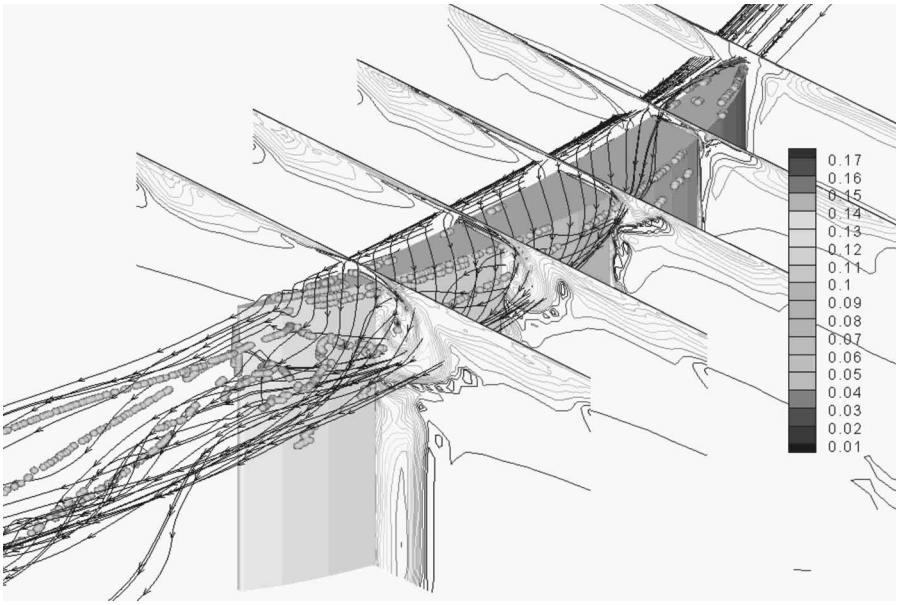

**Fig. 6 Streamlines and flow patterns in the tip gap. (*a*) Experiments [19], (*b*) SUPG+DRDJ, and (*c*) VSGS+DRDJ (Sp: saddle point; PV: passage vortex; TL: tip leakage; HSV: horse-shoe vortex).**

Figure 6 compares the predictions of the secondary flow phenomena developing at the endwall, under the influence of tip leakage vortex. The streamline behaviors predicted are compared with the flow physics as interpreted by Kang and Hirsch [19]. The figure shows the streamline patterns at various significant locations, obtained by the SUPG+DRDJ and VSGS+DRDJ methods. The two solutions are very close and compare well to the experimental interpretation.

Figure 7 shows the chordwise evolution of the normalized turbulent intensity (TI) on cross-flow planes. The TI isolines are visualized with the tip leakage vortex streamlines. When comparing the vortex cores, we see that both numerical formulations predict fairly well the complex multiple-vortex aerodynamics at the tip [19]. They successfully detect (i) the main tip leakage vortex, (ii) a tip separation vortex traveling from the pressure to

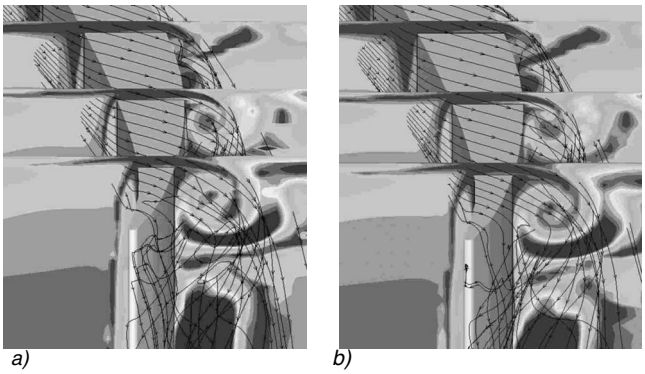



*a*)




*b*)

**Fig. 7 TI distribution and tip leakage vortices detection. (*a*) SUPG+DRDJ and (*b*) VSGS+DRDJ.**



*a)* *b)*

**Fig. 8 Normalized turbulent viscosity $\nu_t/\nu$ distribution. (*a*) SUPG+DRDJ and (*b*) VSGS+DRDJ.**

the suction side of the blade, and (iii) the separation vortex at the leading edge.

Figure 8 shows the turbulent viscosity $\nu_t$ (normalized by the molecular viscosity $\nu$) in the vicinity of the blade tip. The $\nu_t$ contours are shown on the cross-flow planes to describe its axial evolution within the swirling region at the tip. The comparison shows that the VSGS+DRDJ formulation exhibits less diffusive behavior in the region with the largest solution gradient.

## 7 Conclusion

In this paper, we presented a variational multiscale method for turbulence modeling based on the RANS approach. The method addresses the numerical issues related to dominant reactionlike terms involved in the turbulence model. A discontinuity-capturing term provides the additional stabilization needed in parts of the flow domain where the reactionlike terms become large, without

introducing excessive dissipation in other parts of the domain. The test computations presented are for a 2D model problem and 3D flow computation for a linear compressor cascade, and they establish the effectiveness of the method.

## Acknowledgment

The authors acknowledge MIUR support under the projects *Ateneo* and *Visiting Professor Programme* at University of Rome "La Sapienza."

## References

[1] Corsini, A., Rispoli, F., and Santoriello, A., 2004, "A New Stabilized Finite Element Method for Advection-Diffusion-Reaction Equations Using Quadratic Elements," *Modelling Fluid Flow*, T. Lajos, et al., eds., Springer, New York.

[2] Corsini, A., Rispoli, F., and Santoriello, A., 2005, "A Variational Multiscale High-Order Finite Element Formulation for Turbomachinery Flow Computations," Comput. Methods Appl. Mech. Eng., **194**, pp. 4797–4823.

[3] Corsini, A., Rispoli, F., Santoriello, A., and Tezduyar, T. E., 2006, "Improved Discontinuity-Capturing Finite Element Techniques for Reaction Effects in Turbulence Computation," Comput. Mech., **38**, pp. 356–364.

[4] Dubois, T., Jauberteau, F., and Temam, R., 1993, "Solution of the Incompressible Navier–Stokes Equations by the Nonlinear Galerkin Method," J. Sci. Comput., **8**, pp. 167–194.

[5] Hughes, T. J. R., Mazzei, L., and Jansen, K. E., 2000, "Large Eddy Simulation and the Variational Multiscale Method," Comput. Visualization Sci., **3**, pp. 47–59.

[6] Rispoli, F., Corsini, A., and Tezduyar, T. E., 2007, "Finite Element Computation of Turbulent Flows With the Discontinuity-Capturing Directional Dissipation (DCDD)," Comput. Fluids, **36**, pp. 121–126.

[7] Tezduyar, T. E., and Park, Y. J., 1986, "Discontinuity Capturing Finite Element Formulations for Nonlinear Convection-Diffusion-Reaction Equations," Comput. Methods Appl. Mech. Eng., **59**, pp. 307–325.

[8] Tezduyar, T. E., Park, Y. J., and Deans, H. A., 1987, "Finite Element Procedures for Time-Dependent Convection-Diffusion-Reaction Systems," Int. J. Numer. Methods Fluids, **7**, pp. 1013–1033.

[9] Codina, R., 1998, "Comparison of Some Finite Element Methods for Solving the Diffusion-Convection-Reaction Equation," Comput. Methods Appl. Mech. Eng., **156**, pp. 185–210.

[10] Franca, L. P., and Valentin, F., 2000, "On an Improved Unusual Stabilized Finite Element Method for the Advective-Reactive-Diffusive Equation," Comput. Methods Appl. Mech. Eng., **190**, pp. 1785–1800.

[11] Hughes, T. J. R., 1995, "Multiscale Phenomena: Green's Functions, the Dirichlet-to-Neumann Formulation, Subgrid Scale Models, Bubbles and the Origins of Stabilized Methods," Comput. Methods Appl. Mech. Eng., **127**, pp. 387–401.

[12] Hauke, G., 2002, "A Simple Subgrid Scale Stabilized Method for the Advection-Diffusion-Reaction Equation," Comput. Methods Appl. Mech. Eng., **191**, pp. 2925–2947.

[13] Gravemeier, V., and Wall, W. A., 2007, "A 'Divide-and-Conquer' Spatial and Temporal Multiscale Method for Transient Convection-Diffusion-Reaction Equations," Int. J. Numer. Methods Fluids, **54**, pp. 779–804.

[14] Craft, T. J., Launder, B.E., and Suga, K., 1996, "Development and Application of a Cubic Eddy-Viscosity Model of Turbulence," Int. J. Heat Fluid Flow, **17**, pp. 108–155.

[15] Corsini, A., and Rispoli, F., 2005, "Flow Analyses in a High-Pressure Axial Ventilation Fan With a Non-Linear Eddy Viscosity Closure," Int. J. Heat Fluid Flow, **17**, pp. 108–155.

[16] Tezduyar, T. E., 2003, "Computation of Moving Boundaries and Interfaces and Stabilization Parameters," Int. J. Numer. Methods Fluids, **43**, pp. 555–575.

[17] Tezduyar, T. E., 2007, "Finite Elements in Fluids: Stabilized Formulations and Moving Boundaries and Interfaces," Comput. Fluids, **36**, pp. 191–206.

[18] Tezduyar, T. E., and Osawa, Y., 2001, "Finite Element Stabilization Parameters Computed From Element Matrices and Vectors," Comput. Methods Appl. Mech. Eng., **190**, pp. 411–430.

[19] Kang, S., and Hirsch, C., 1993, "Experimental Study on the Three-Dimensional Flow Within a Compressor Cascade With Tip Clearance: Part I—Velocity and Pressure Fields," ASME J. Turbomach., **115**, pp. 435–443.

[20] Kang, S., and Hirsch, C., 1993, "Experimental Study on the Three-Dimensional Flow Within a Compressor Cascade With Tip Clearance: Part II—The Tip Leakage Vortex," ASME J. Turbomach., **115**, pp. 44–452.